1967-09

The analysis and design of communication nets using reliability functions.

Huber, Raymond James

Monterey, California. U.S. Naval Postgraduate School

THE ANALYSIS AND DESIGN OF COMMUNICATION NETS USING RELIABILITY FUNCTIONS

RAMOND JAMES HUBER

THE ANALYSIS AND DESIGN OF

COMMUNICATION NETS

USING RELIABILITY FUNCTIONS

by

Raymond James Huber
Lieutenant, United States Navy
B.S., Franklin and Marshall College, 1960

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1967

ABSTRACT

Topological methods may be employed in an effective manner in the analysis and design of a communication net. One application is in the determination of the reliability of the net with respect to both complete communication and k-terminal communication. The effect on this reliability caused by the removal of one link is studied.

The k-terminal reliability function is defined and its application to analysis and design of communication nets is demonstrated. A digital computer program is presented, and examples of its use are included. A formula is given to find the value of the 2-terminal function.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENT

The author is indebted to Dr. Shu-Gar Chan of the Naval Postgraduate School for his guidance and direction provided during the preparation of this thesis.

1. Introduction.

Any system that provides for communication, be it the transmission of information or the transportation of commodities, is subject to possible failure. The loss of security, the malfunction of a radio transmitter, and the destruction of a bridge are examples of this failure. Linear graph theory, as described in texts such as [5], has been applied to measure the reliability of the system given some measure of the reliability of the parts. Fu and Yau, [4], have indicated a value for the probability that two stations can communicate; and a digital computer algorithm for this has been given, [6]. The probability that all pairs of stations can communicate within a given time interval has been studied by Chan, [2], and Fu, [3].

In this work the application of the k-terminal reliability function, to be defined in Section 3, is presented, and a digital computer program is included. In addition, the application of the power of this function without computer assistance is demonstrated. A formula is given to find the value of the 2-terminal reliability function without enumerating the Table of Combinations.

2. Statement of the Problem.

A communication net may be represented by a linear graph in which the edges of the graph represent the communication links, and the nodes of the graph represent the stations of the communication net.

A weight can be assigned to each edge of the graph to indicate the probability that the particular communication link represented by the edge will operate successfully. This weight is called the reliability of that edge. This probability is not a conditional probability. Each edge is assumed to be associated with an independent random variable, having only two possible states: normal operation, and not operational.

The communication net can be analyzed by studying the topology of the graph. Topological methods can be applied on several different problems. For example, if one link of a communication net has been lost (destroyed), it must be decided whether or not to replace this link with one of the other existing links in the net. The information needed to make this decision must include the relative importance of each link, some measure of the quality of the branch (e.g., its reliability), and the role of the branch in the net. The analysis of the data must include the degree of importance attached to the overall net reliability as contrasted to the reliability of communication between a pair or a group of terminals.

Several investigators, [3], [4], [6], have proposed quantitative measures for the reliability of a communication net. Investigations by Chan [2], have led to a result which uses the reliability function as a k-terminal reliability function. This function may be applied to the reliability of communication between any k-number of nodes. The topological methods which are used to generate the function are suitable

for use on a digital computer. The large storage requirement is the limiting factor in their application.

3. Properties of Reliability Functions.

3.1 2-terminal Reliability Function.

The reliability function is defined in reference [2]. Some of the most important definitions are included here for ready reference and emphasis. Using the notation of [2], there is a value of reliability, $p_1, p_2, \ldots p_e$, assigned to each of the edges of the graph, $e_1, e_2, \ldots e_k, \ldots e_e$. Each branch of the net is assumed to be associated with an independent random variable. If the branch, b_i, is operational, the edge, e_i, of the graph assumes the value p_i, which is the probability that edge, e_i, will operate normally. If the branch is not operational, the edge, e_i, assumes the value $\bar{p}_i$, which is the probability that the edge, e_i, will not be operational. Although these reliabilities are probabilities, they are considered not to be conditional probabilities. That is, we have the following relations:

$$p_i + \bar{p}_i = 1 \tag{3.1}$$

$$\Pr(A \text{ and } B) = \Pr(A)\Pr(B) \tag{3.2}$$

Definition 3.1: 2-terminal Reliability Function:
A reliability function expressed in terms of the edge reliabilities of a network is a 2-terminal reliability function if it represents the reliability between two distinct terminals (i.e., the probability of communication between these two terminals).

Let $q_{x,y}$ represent the 2-terminal reliability function between terminals x and y. Then we have

$$q_{x,y} = m_1\bar{p}_1\bar{p}_2\cdots\bar{p}_e + m_2\bar{p}_1\bar{p}_2\cdots\bar{p}_{e-1}p_e + \ldots + m_{2^e}p_1p_2\cdots p_e \tag{3.3}$$

where $m_1, m_2, \ldots m_i, \ldots m_{2e}$ are either 1 or 0 depending on whether the combination of states of the edges provide a communication path between terminals $\underline{x}$ and $\underline{y}$. If the path exists, then $m_i = 1$; if not, $m_i = 0$. When the function is expressed as in equation (3.3), it is said to be in its canonical form.

Definition 3.2: Path Product.

A path set is a set of branches of a communication path between a pair of terminals. A path product is the product of the variables associated with the elements in a path set.

Definition 3.3: Path Product Term or Primary Path Product Term.

A canonical term is a path product term if

(1) its unbarred variables are those of a path product;and

(2) all other variables in the term are barred variables.

Definition 3.4: Secondary Path Product Term.

A canonical term is a secondary path product term if its unbarred variables correspond to those elements forming a path set plus one or more not in the path set.

Definition 3.5: Table of Combinations.

The collection of all nonzero canonical terms of a 2-terminal reliability function is called the Table of Combinations. That is, all the terms in which m_i takes on the value 1.

Definition 3.6: k-edged Path.

A k-edged path is a path of $\underline{k}$ number of edges or branches which constitute a communication path between a pair of terminals.

Theorem 3.1:

Let P_A be an A-edged path between terminals $\underline{x}$ and $\underline{y}$, in a graph containing $\underline{e}$ number of edges. The Table of Combinations for the 2-terminal reliability function, $q_{x,y}$, will include a primary path product term for path P_A, and $2^{(e-A)}-1$ secondary path terms.

Proof:

From definition (3.3) and (3.5), the primary path product term must be present. There are $2^{(e-A)}$ possible combinations of the (e-A) variables not specified by the primary term. However, definition (3.4) requires the number of unbarred variables to be at least (A+1). Therefore, the maximum possible number of combinations is $2^{(e-A)}-1$.

If a graph contains e number of edges and n number of vertices, then definition (3.1) indicates that there will be 2^e number of terms in the 2-terminal reliability function. Theorem (3.1) indicates that one path of A number of edges will produce $2^{(e-A)}$ terms in the Table of Combinations. This is the same as saying that, from definition (3.1), m_i will be nonzero in $2^{(e-A)}$ terms of the 2-terminal reliability function.

Lemma 3.1:

If no path exists between two terminals x and y, then the 2-terminal reliability function will have no nonzero terms.

Proof:

By definition (3.1) all the m_i values will be zero, since there is no path between terminals x and y. There will be no entries in the Table of Combinations.

Theorem 3.2:

If a graph contains e number of edges and n number of vertices, and if an A-edged path is the only path between vertices x and y, then there will be $2^{(e-A)}$ number of terms in the 2-terminal reliability function, $q_{x,y}$.

Proof:

By theorem (3.1) there will be $2^{(e-A)}$ number of terms for the A-edged path. Since there are no more paths, lemma (3.1) shows there will be no additional terms. Hence, the theorem.

Theorem 3.3:

Let P_A be an A-edged path and P_B a B-edged path, each between terminals $\underline{x}$ and $\underline{y}$, in a graph containing $\underline{e}$ number of edges. The primary path product term of path set A can not be a secondary path product term for path set B in $q_{x,y}$.

Proof:

The unbarred elements in the secondary path product terms must appear as barred elements in the primary path product term of the path set A, by definition (3.3). Hence, the theorem.

Theorem 3.4:

Let P_A and P_B be path sets between terminals $\underline{x}$ and $\underline{y}$. Then at least one secondary path product term for path set A will be identical to one for path set B.

Proof:

From definition (3.4), the collection of secondary path product terms for any path set will include the case where all variables are unbarred.

This will be common to all collections of secondary path product terms. Hence, there will always be this term present.

Example 3.1:

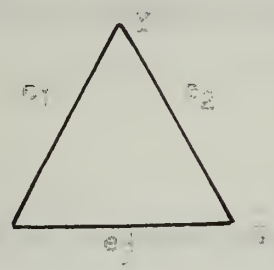


Figure 3.1

Path Set A $P_A = \{e_1\}$

Path Set B $P_B = \{e_2 e_3\}$

2-terminal reliability function, $q_{x,y}$, expressed in canonical form:

$$q_{x,y} = m_1\bar{p}_1\bar{p}_2\bar{p}_3 + \quad (7)\ (10)$$

$$m_2\bar{p}_1\bar{p}_2p_3 + \quad (8)$$

$$m_3\bar{p}_1p_2\bar{p}_3 + \quad (9)$$

$$m_4\bar{p}_1p_2p_3 + \quad (2) \quad (11)$$

$$m_5p_1\bar{p}_2\bar{p}_3 + \quad (1)$$

$$m_6p_1\bar{p}_2p_3 + \quad (3) \quad (12)$$

$$m_7p_1p_2\bar{p}_3 + \quad (4) \quad (13)$$

$$m_8p_1p_2p_3 \quad (5)\ (6) \quad (14)$$

(1) primary path product term for path set A

(2) primary path product term for path set B

(3)(4)(5) secondary path product terms for path set A

(6) secondary path product term for path set B

observe that (5) and (6) are identical

(7)(8)(9) m_i terms are 0 because terms do not contain a path between terminals x and y

(10) this term will always be zero

(11)(12)(13) a tree of the graph is made up of the unbarred terms

(14) a tree plus a chord of the graph is made up by the unbarred terms

3.2 Multi-terminal Reliability Function.

Reliability functions, which are expressed in terms of the edge reliabilities, may be derived in order to represent the reliability of transmission between more than two terminals. Thinking of a communication net, one can envision a classification of the terminals into the categories of transmitter, receiver, or both, and repeaters. The repeaters may be looked upon as intermediate steps in the communication paths. All of this may enter into an evaluation of the reliability of simultaneous communication between all pairs of terminals.

Definition 3.7: Complete Communication.

The event that every pair of terminals x and y in a net can communicate with each other simultaneously is complete communication; the simultaneous communication between all pairs of terminals of the net.

Definition 3.8: N-terminal Reliability.

The probability of complete communication in a net with N-terminals is called the "N-terminal reliability" of the net and is denoted by q_n.

Definition 3.9: k-terminal Reliability.

In a set of n number of terminals, consider k number of terminals of transmission, where $k < n$. Then, the probability of complete communication among the set of k number of terminals is called the k-terminal reliability of the net and is denoted by q_k.

From definitions (3.7) and (3.9), a net may have several different values of k-terminal reliability depending on which terminals are included in the set of k number of terminals of transmission. The

choice of terminals included in this set is of fundamental importance in the interpretation of the analysis or data collection.

For a given net, q_n is found in the following way: first determine $q_{x,y}$ for each pair of vertices. There will be $\frac{1}{2}(n)(n-1)$ for a net with n number of terminals. Then sum all the canonical terms which are common to all $q_{x,y}$ functions. To find q_k, find $q_{x,y}$ for each pair of terminals in the set, K. There will be $\frac{1}{2}(k)(k-1)$ number. Then sum all the canonical terms common to all these $q_{x,y}$ functions.

Another method is first to determine a complete set of trees for the net. Complete communication is possible if and only if the unbarred variables of a canonical term include those corresponding to the branches of a tree. By taking the sum of all canonical terms that contain a tree, or a tree plus chords, made up from the unbarred variables, q_n is determined. If the terms which are summed contain a subtree of the graph, or a subtree plus chords, containing all the vertices of the set K, the q_k is determined. This method can be applied on the canonical form resulting after the first $q_{x,y}$ calculation. This can reduce the labor required to find q_n or q_k provided all the trees of the graph are known.

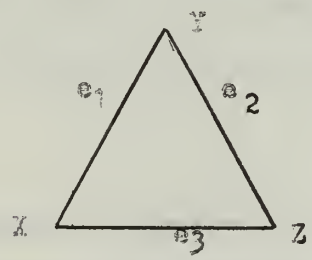


Figure 3.1 repeated

Path set $P_C = \{e_2\}$

Path set $P_D = \{e_1 e_3\}$

Path set $P_E = \{e_3\}$

Path set $P_F = \{e_1 e_2\}$

Part A

$$q_{y,z} = m_1\bar{p}_1\bar{p}_2\bar{p}_3 + m_2\bar{p}_1\bar{p}_2p_3 + m_3\bar{p}_1p_2\bar{p}_3 + (1) \quad m_4\bar{p}_1p_2p_3 + (3)\ (7)$$

Part B

$$q_{x,z} = m_1\bar{p}_1\bar{p}_2\bar{p}_3 + m_2\bar{p}_1\bar{p}_2p_3 + (1) \quad m_3\bar{p}_1p_2\bar{p}_3 + m_4\bar{p}_1p_2p_3 + (3)\ (7)$$

$m_5 p_1 \bar{p}_2 \bar{p}_3$ +

$m_6 p_1 \bar{p}_2 p_3$ + (2) (8)

$m_7 p_1 p_2 \bar{p}_3$ + (4) (9)

$m_8 p_1 p_2 p_3$ (5) (6) (10)

$m_5 p_1 \bar{p}_2 \bar{p}_3$ +

$m_6 p_1 \bar{p}_2 p_3$ + (4) (8)

$m_7 p_1 p_2 \bar{p}_3$ + (2) (9)

$m_8 p_1 p_2 p_3$ (5) (6) (10)

(1) Primary path product term for path set C and E.

(2) Primary path product term for path set D and F.

(3)(4)(5) Secondary path product term for path set C and E.

(6) Secondary path product term for path set D and F.

(7) (8)(9) (10) make up a tree, or a tree plus a chord, of the graph, made up of the unbarred terms.

Example 3.3

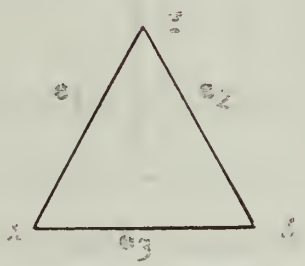


Figure 3.1 repeated

Number of trees, t

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = 3$$

The 3 trees of the graph are:

$$\{e_1 e_2\} \ , \quad \{e_2 e_3\} \ , \quad \{e_1 e_3\}$$

Using the results of examples (3.2) and (3.1), we have

$$q_n = m_4 \bar{p}_1 p_2 p_3 + m_6 p_1 \bar{p}_2 p_3 + m_7 p_1 p_2 \bar{p}_3 + m_8 p_1 p_2 p_3$$

by the process of summing the canonical terms in common. From example (3.1) alone, or (3.2) part (a) alone or part (b) alone, the same results

could have been obtained using the method of trees.

In this simple example, for any set of terminals of transmission less than the given set, the problem reduces to one of the examples, since it becomes a 2-terminal reliability problem.

3.2 Some Characteristics of the Path Product Terms.

A graph contains 5 edges. A matrix can be constructed in which the elments are the number of path product terms that contain U number of unbarred variables compared to the L number of edges in the path.

Number of unbarred variables

Path Length	1	2	3	4	5
4	0	0	0	1	1
3	0	0	1	2	1
2	0	1	3	3	1
1	1	4	6	4	1

The number of path product terms

This array is recognized as the start of Pascal's Triangle. See reference [1]. The nonzero matrix elements are the so called binary coefficients, defined by the formula

$$\binom{n}{j} = \frac{n!}{j!(n-j)!} \tag{3.4}$$

where, in this case, n = (e-A), and A is the path length, and j = 0, 1,2,..., (e-A).

If all the branches of the net have the same reliability, p, then the collection of all path product terms resulting from an A-edged path, in a graph with e edges, can be written as follows:

$$\binom{e-A}{0} p^{A}\bar{p}^{(e-A)} + \binom{e-A}{1} p^{(A+1)}\bar{p}^{(e-A-1)} + \binom{e-A}{2} p^{(A+2)}\bar{p}^{(e-A-2)} +$$

$$\cdots + \binom{e-A}{e-A-1} p^{(e-1)}\bar{p}^{(1)} + \binom{e-A}{e-A} p^{e} = \sum_{i=0}^{e-A} K(i) p^{(A+i)}\bar{p}^{(e-A-i)} \qquad (3.5)$$

where K(i) is the collection of the binary coefficients defined by the formula

$$\binom{e-A}{i} = \frac{n!}{i!(n-i)!}$$

Examples (3.1) and (3.2) illustrate the generation of the primary and secondary path product terms. Theorem (3.4) points out that there will always be at least one of the secondary path product terms common to the expansions generated from two or more path sets is a 2-terminal reliability function. Theorem (3.3) proves that the primary path product term is never a secondary path product term for another path in the 2-terminal reliability function; that is, it is not one of the duplicate terms. The exact number of unique terms added to this function is a problem in conditional probability. The following group of theorems is fundamental to the solution of this problem.

The theorems and lemmae in this section are subjected to the following conditions: a graph contains e number of edges and n number of vertices. Path set, $P_{1, xy}$, is a 1-edged path set between terminals x and y, made up of branch b, alone.

Theorem 3.5:

No multi-edged path between terminals x and y may contain branch b.

Proof:

By definition (3.2) a path set is a set of branches of a communication path between a pair of termunals. No additional branches are included in the path set. Hence, since branch b is a path by itself, any set that is made up of branch b and one or more additional branches cannot be a path set between terminals x and y. Definition (3.6) of a k-edged path allows the conclusion that no multi-edged path between terminals x and y contain branch b.

Lemma 3.2:

There can be no primary path product term in which branch b appears as an unbarred variable in the 2-terminal reliability function, $q_{x,y}$, except path set $P_{1,xy}$.

Proof:

Lemma (3.2) follows immediately from theorem (3.5) and definition (3.3).

Lemma 3.3:

There can be no secondary path product terms generated from any path between x and y in which branch b appears as an unbarred variable except path set $P_{1,xy}$.

Proof:

This lemma follows directly from theorem (3.5) and definition (3.4).

Theorem 3.6:

Let path set, $P_{2,xy}$, be an A-edged path set between terminals $\underline{x}$ and $\underline{y}$. This path will generate $2^{(e-A-1)}$ secondary terms that are duplicates of those generated by $P_{1,xy}$.

Proof:

The number of possible unique combinations of the (e-A) number of barred variables has decreased by a factor of 2 because the 1-edged path, $P_{1,xy}$, has previously formed $2^{(e-1)}$ combinations of the same variables. Only those in which the variable $\underline{b}$ appears as a barred variable remain ungenerated. That number is $2^{(e-A-1)}$. There were $2^{(e-A)}$ possible, but now only $2^{(e-A-1)}$, meaning that $2^{(e-A-1)}$ were duplicates.

Investigations have shown that the number of duplicates generated among all the path product terms are arranged in correspondence with the binary coefficients. That is, if there are eight duplicated generated, there will be a 1-3-3-1 distribution, based on the number of unbarred variables in the path. If there are D number of duplicates, then the distribution will be as given by

$$\binom{D}{j} = \frac{D!}{j!\ (D-j)} \quad , \ j = 0,\ 1,2,\ldots,D$$

The number of unique path product terms that are generated by each path set, P_i, in the 2-terminal reliability function, $q_{x,y}$, can be found by the following method.

a) Write the path matrix, order (p x e), where $\underline{p}$ is the number of path sets between terminals $\underline{x}$ and $\underline{y}$, and $\underline{e}$ number of edges in the graph.

b) Rearrange the rows such that the shortest path is row #1, the next shortest is row #2, and so forth, until the last

row contains the longest path.

c) Count the number of columns in the row in which there is no entry in any of the preceeding rows. For example, this number will be (e-k) for the first row. This number may be zero. Call this number n_i.

d) Calculate 2^{n_i}. This result is the number of unique path product terms generated by the path set in the 2-terminal reliability function.

The total number of terms in the Table of Combinations is the sum of the results of step (d) above. It is seen that if n_i is zero, that one term is still added to the Table of Combinations by the path, which agrees with Theorem (3.3). If there is only one path, then there will be $2^{(e-k)}$ number of terms which agrees with Theorem (3.2). Further statements which tend to prove the validity of this method follow the presentation of this example.

Example 3.4:

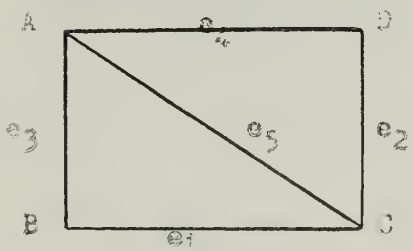


Figure 3.2

The paths between vertices A and B:

$$\{e_1e_5\}\ ,\ \{e_3\}\ ,\{e_1e_2e_4\}$$

Path Matrix:

edges:	1	2	3	4	5
paths: 1	1	0	0	0	1
2	0	0	1	0	0
3	1	1	0	1	0

After rearranging, the path matrix becomes:

edges:		1	2	3	4	5
paths:	2	0	0	1	0	0
	1	1	0	0	0	1
	3	1	1	0	1	0

Counting the number of columns in each row in which there is no entry in that column in any of the preceeding rows:

		n_i:
paths:	2	4
	1	2
	3	0

Calculating:

		2^{n_i}:
paths:	2	16
	1	4
	3	1

From definitions (3.1 - 5), there are established conditions which a collection of variables must meet in order to be an entry in the Table of Combinations. Looking at example (3.4), one can see that path (2) will generate 2^4 unique terms. After they have been generated, path (1) generates a collection of 2^3 number of possible terms. All these in which edge (3) appears as an unbarred variable will be duplicates of those generated by path (1). This reduces the number of terms by a factor of 2, yielding the result: 2^2. This can be understood from the concept of combinations; there were only 2 elements to be combined in 2 different ways vice what had appeared to be 3 elements; i.e., 2^2 vice 2^3.

The same argument holds for path (3). While there would appear to be 2 elements that can be varied, there are none which have not already been varied. The fact that the primary path product term is unique itself is given in theorem (3.3). It is from this logic that the above stated procedure is derived.

The knowledge of these numbers leads to a method to determine the value of the 2-terminal reliability function. With equation (3.5) in mind, a new equation can be written for each path set using the numbers, n_i, found by the above described procedure. Let $P_{x,y}$ be a k-edged path between terminals x and y in a graph with e number of edges. For this path, let $n_i = N$. Then one can write

$$(r_i r_j \ldots r_k)\ (\bar{r}_L \bar{r}_M \ldots \bar{r}_N) \left\{ \binom{N}{0} \bar{r}^N + \binom{N}{1} r\bar{r}^{(N-1)} + \ldots + \binom{N}{N-1} r^{(N-1)} \bar{r} + \binom{N}{N} r^N \right\} = \left| Q \right|_{P_k} \tag{3.6}$$

where $\left| Q \right|_{P_k}$ is the value of all the path product terms added to the Table of Combinations by the path, P_k.

Where the first factor is the product of the k number of branch reliabilities associated with the branches in the path, $P_{x,y}$; the second factor is the product of the (e-k-N) number of complements of the branch reliabilities of the branches which have previously appeared as unbarred variables in other path product terms; the third factor is the summation of all the path product terms in which N number of variables, associated with the branches not in the path, $P_{x,y}$, and not previously specified as being unbarred in other path product terms, appear as barred and unbarred elements.

However, the quantity inside the brackets is equal to one. This statement is true because the quantity inside the brackets is an application of the binomial theorem for two conditions: the branch reliabilities being all equal; and the branch reliabilities being not all equal.

Case 1) Let all the branch reliabilities be equal to r. Then the quantity inside the brackets of equation (3.6) becomes

$$(r + \bar{r})^N$$

Since $(r + \bar{r}) = 1$, then $(r + \bar{r})^N = 1$.

Case 2) Let the reliability of branch, b_i, be given by r_i. Then the quantity inside the brackets of equation (3.6) may be rewritten:

$$(r_1 + \bar{r}_1)(r_2 + \bar{r}_2) \ldots (r_N + \bar{r}_N) = (1)(1)\ldots(1) = 1$$

Therefore, equation (3.6) can be rewritten

$$(r_i r_j \ldots r_k)(\bar{r}_L \bar{r}_M \ldots \bar{r}_N) = \left|Q\right|_{P_k} \tag{3.7}$$

In the event all branch reliabilities are equal,

$$r^{k}\bar{r}^{(e-k-N)} = \left|Q\right|_{P_k} \tag{3.8}$$

4. Topological Analysis of the Net.

4.1 Collection of All Possible Path Sets.

Fundamental to the analysis of any communication net is the knowledge of all possible paths between the terminals x and y. Since no path may be ignored, even in relatively small, simple nets, the task of finding all paths is a laborous task. Linear graph theory provides an efficient method to accomplish this task.

Consider a graph with e number of edges and n number of vertices, and define a circuit to be a closed edge train. Then, the theory of topology has shown there to be (e - n + 1) = w number of independnet circuits in the graph. These are called the fundamental circuit sets, and are denoted by B_f. Any additional circuits that appear to exist in the graph can be shown to be algebraically derived from the set of independent circuits.

Next there must be built a collection of sets, called set B, which includes the null set, the W number of sets of B_f, and the collection of all possible combinations of circuit sets, which are generated by the ringsum operation of two or more circuit sets in B_f. The ringsum operation is defined as modulo 2 addition of the respective column entries of the two sets. (See example, (4.2). The set B will contain 2^w number of sets.

A path set, $P_{x,y}$, is chosen, and expressed as a row matrix of e columns. Set B is represented in matrix form of order (w x e). The ringsum operation of $P_{x,y}$ and each set in B produces a new set which is written in a collection of sets called set BRP. This is also in matrix form of order (w x e). Linear graph theory shows that the ringsum of a path set and a circuit set will be either a new path set or the disjoint

union of a path set and a circuit set. Now each entry in BRP must be examined to retain only the pure path sets, which are written in set PFC, also in matrix form of order (p x e), where p is the number of paths between x and y.

One process of examining set BRP is to look at each entry first to see whether the original path set, $P_{x,y}$, remains. If it does, and if there are other branches present, then it is not a pure path set and must be discarded. If it does not, then it must be ascertained whether any of the circuits of set B are included. If so, then it is not a pure path set. Those that remain are the pure path sets, and more importantly, all the path sets between terminals x and y. A block diagram of this procedure is given in figure (4.2).

The ringsum operation is built into the computer program by means of three FORTRAN IV instructions. The block diagram for this operation is shown in figure (4.3). Example (4.1) shows the ringsum operation for the four possible cases. The parameters must be logical elements; that is, either True, or False, with a 1 bit representing True, and a 0 bit False.

Example 4.1:

	Case 1	Case 2	Case 3	Case 4
A =	0	1	1	0
B =	1	0	1	0
C =	1	1	0	0

The step-by-step computer program solution proceeds as follows:

after step 1, C =	0	0	0	0
after step 2, C =	1	1	1	0
after step 3, C =	1	1	0	0

The final results are identical.

Example 4.2:

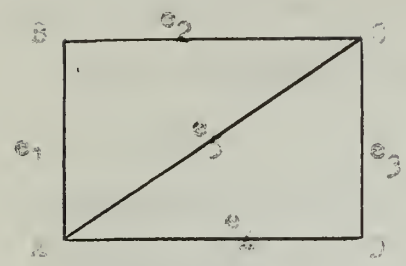


Figure 4.1

$e = 5$

$m = 4$

$w = 5 - 4 + 1 = 2$

$2^w = 2^2 = 4$

Circuit (1) $\{e_1\ e_2\ e_5\}$

Circuit (2) $\{e_3\ e_4\ e_5\}$

Set B_f:

edge number		1	2	3	4	5
Circuit number	1	1	1	0	0	1
	2	0	0	1	1	1

Order ($w \times e = 2 \times 5$)

Circuit (3) = $\{e_1e_2e_3e_4\}$ is derived algebrically by ringsum of circuit (1) and (2) as follows

1	1	0	0	1
0	0	1	1	1
1	1	1	1	0

Choose $P_{A,B} = \{e_1\}$

Set B:

edge number		1	2	3	4	5
set number	1	0	0	0	0	0
	2	1	1	0	0	1
	3	0	0	1	1	1
	4	1	1	1	1	0

Order ($2^w \times e = 4 \times 5$)

$P_{AB} \oplus B = BRP$

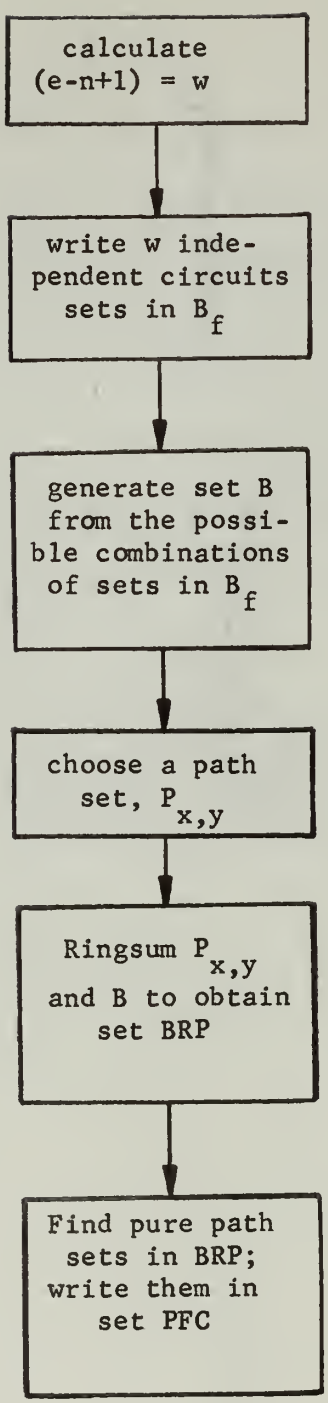
Set BRP =

edge number		1	2	3	4	5	
set number	1	1	0	0	0	0	
	2	0	1	0	0	1	
	3	1	0	1	1	1	
	4	0	1	1	1	0	Order (2^w x e = 4 x 5)

Set PFC =

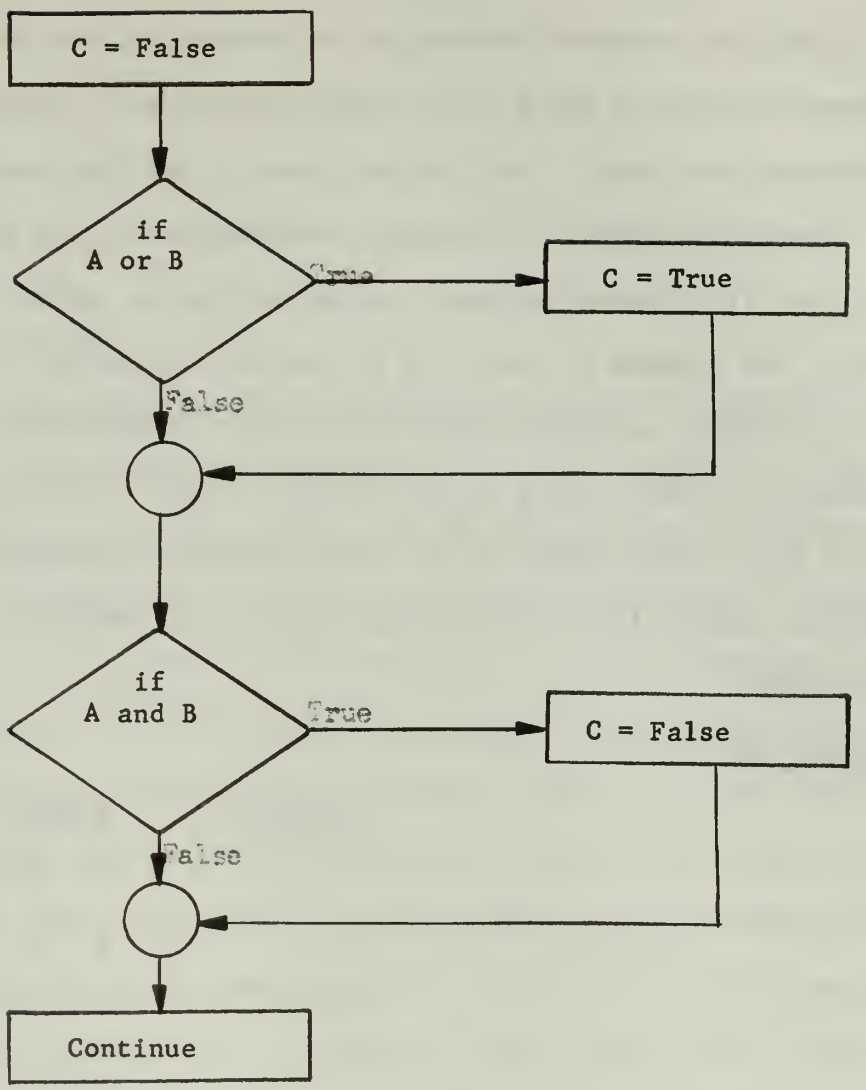
edge number		1	2	3	4	5	
path number	1	1	0	0	0	0	
	2	0	1	0	0	1	
	3	0	1	1	1	0	Order (p x e = 3 x 5)

Paths are $\{e_1\}$, $\{e_2\ e_5\}$, $\{e_2 e_3 e_4\}$.



Block Diagram of Process to Find All Paths

Figure 4.2



Block Diagram of Ringsum Operation in Computer Program

Figure 4.3

4.2 Path Finding Program.

A digital computer program can be devised to find all the paths between terminals x and y. The program discussed in Appendix (6) accomplishes this task. (For further details, see that section).

The generation of all possible combinations of the fundamental circuit sets is an important step, and no one may be omitted. The method used by the program to find them all is the following: to count from 4 to 7 in binary, one adds "100" to the binary numbers from 0 through 3. Similarly to count from 8 to 15, one adds "1000" to the binary numbers from 0 to 7. See example (4.2). This procedure guarantees that all possible combinations are obtained, because all numbers are obtained in the analogy.

Example 4.3:

Decimal	Binary
0 =	0 0 0 0
1 =	0 0 0 1
2 =	0 0 1 0
3 =	0 0 1 1

to count from 4 to 7 in binary, add 100 to those numbers preceeding it.

4 =	0 1 0 0
5 =	0 1 0 1
6 =	0 1 1 0
7 =	0 1 1 1

Similarly, from 8 to 15, add 1000

8 =	1 0 0 0
9 =	1 0 0 1
10 =	1 0 1 0
11 =	1 0 1 1

Decimal	Binary
12 =	1 1 0 0
13 =	1 1 0 1
14 =	1 1 1 0
15 =	1 1 1 1

4.3 Branch Loading.

When a communication net is analyzed, each link is examined individually. In addition to the reliability of the link to perform its function, there is needed a measure of the relative importance of the link in the various modes of operation of the complete net. For example, the importance of a link may be different, if complete communication is required, than if only K-terminal communication is required. One such measure of relative importance of a link is called the load of a branch.

Fu has defined the load of a branch to be the total number trees that contain this branch, [3]. As shown before, complete communication is possible only when the net contains at least a tree in the operational state. Since each branch is contained in one or more trees of the net, the number of trees that contain this branch can be found, and this number used to indicate the relative importance of the branch in complete communication. This number can be normalized, if desired, since the total load of all branches can be quickly calculated by a formula shown by Fu, [3], as

$$\sum_{i=1}^{n} l_i = (v - 1)t$$

where l_i (i = 1, 2, ...,m) is the load of branch b_i in a network with v number of vertices, n branches, and t number of trees.

This method of branch loading determination can be extended to the k-terminal communication case by means of subtrees. However, branch loading can also be defined in another way.

Definition 4.1: Branch Loading.

The load of a branch, b_i, is defined to be the total number of paths (path sets) that contain this branch.

A path set was defined to be a set of branches of a communication path between a pair of terminals. When complete communication is considered, definition (4.1) makes use of all the paths between all pairs of terminals. This means that the numbers obtained for the various branch loadings will be larger than those obtained using the definition of Fu; however, the relative magnitudes will be the same. That is, the branch with the highest loading will be identified to be the same by either definition. The advantage of definition (4.1) lies in the simple way to evaluate rather than having to find the trees of the net.

4.4 Path Length

When a communication net is analyzed, all the paths between terminal pairs are found. The loading of the branches is measured. Now the paths themselves must be examined. The length of the communication path is of great significance. We know from experience that the more often data is handled, the greater the chance for error. This is confirmed by this simple example.

Example 4.4:

Let p be the reliability of all branches, b_i, and $p < 1$. Then the reliability through one branch is p. The reliability of communication through two branches is $(p \times p) = p^2$; through j number of branches:

p^j. Since $p < 1$, then $p^j < p$ for all $j > 1$.

This effect of path length appears directly in the 2-terminal reliability function by means of the secondary path product terms. Section (4.6) discusses the effect of the secondary path product terms. See also section (5), Application of Ideas without the Aid of a Computer.

4.5 Branch Reliability.

Much work is currently being done by many investigators in the fields of Operations Research, Reliability, Non-Destructive Testing, and many others, to provide a high quality measurement of the reliability of a communication link. This work does not concern itself with the finding of this value of reliability. It must be pointed out, however, the significance of accuracy in this value and its effect on the interpretation of the results.

Results have shown that considering a net with $\underline{e} = 10$ branches, and assuming all branch reliabilities to be equal to .90, the fourth significant figure has risen to the importance level of being the deciding factor in net configuration.

4.6 Secondary Path Product Terms.

The secondary path product terms of definition (3.4) are not of secondary importance in the determination of the reliability function. As can be seen from example (3.1), the majority of nonzero canonical terms are secondary path product terms. The exact number of these terms was given in theorem (3.1), namely, for an A-edged path, there will be $2^{(e-A)}-1$ number of secondary path product terms.

The theorems in this section are subjected to the following conditions: A graph contains e number of edges, and n number of vertices. The branch reliability, $p_{\_}$ of branch b_i, is greater than its complement,

$\bar{p}_i$, and is the same for all branches.

Definition 4.2: Value of a path product term.

The value of a path product term is defined to be the product of the branch reliabilities of the unbarred variables and the complement of the branch reliabilities of the barred variables of the path product term. It is applicable to both primary and secondary path product terms. It always takes on a value less than one, since it is a product of numbers each less than one.

Theorem 4.1:

Any one of the secondary path product terms has a larger value than the primary path product term associated with it, for any A-edged path, $P_{x,y}$.

Proof:

The value of a primary path product term is determined using definitions (3.3) and (4.2) to be

$$p^A \cdot \bar{p}^{(e-A)}$$

By definition (3.4) a secondary path product term must have at least one more unbarred variable than the primary term. Therefore, we write

$$p^{(A + a)} \cdot \bar{p}^{(e-A-a)}$$

where a is a positive integer, i.e., $a \geq 1$

If $p^A \cdot \bar{p}^{(e-A)} < p^{(A + a)} \cdot \bar{p}^{(e-A-a)}$

Then $1 < p^a \cdot \bar{p}^{(-a)}$

Since $\bar{p} < p$,

Thus $p^A \cdot \bar{p}^{(e-A)} < p^{(A + a)} \cdot \bar{p}^{(e-A-a)}$.

Corollary 4.1:

The sum of the values of the primary and all the secondary path product terms of a 1-edged path is greater than that of any multi-edged path between the same terminals x and y.

Proof:

This Corollary follows directly from the theorem (4.1) by letting A be equal to 1.

Theorem 4.2:

The sum of the values of all the primary and all the secondary path product terms of an A-edged path is greater than that of an (A+a)-edged path, where a is any positive integer; i.e., $a \geq 1$.

Proof:

Equation (3.5) can be written for an (A+a)-edged path.

$$\sum_{i=0}^{e-A-a} K(i)p^{(A+a+i)}\bar{p}^{(e-A-a-i)} = \binom{e-A-a}{0} p^{(A+a)}\bar{p}^{(e-A-a)} +$$

$$\binom{e-A-a}{1} p^{(A+a+1)}\bar{p}^{(e-A-a-1)} + \binom{e-A-a}{2} p^{(A+a+2)}\bar{p}^{(e-A-a-2)} + \cdots \quad (4.1)$$

$$+ \binom{e-A-a}{e-A-a-1} p^{(e-1)}\bar{p}^{1} + \binom{e-A-a}{e-A-a} p^{e}$$

Comparing equation (4.1) to equation (3.5), we see that theorem (4.2) can be proved if it can be shown that, using definition (4.2), the value of equation (3.5) is greater than the value of equation (4.1). This can be done by showing that each coefficient of equation (4.1) is less than that of equation (3.5) for the corresponding power of p.

Consider the general term, i = j. If

$$\binom{e\text{-}A\text{-}a}{e\text{-}A\text{-}a\text{-}j} p^{(e-j)-j} < \binom{e\text{-}A}{a\text{-}A\text{-}j} p^{(e-j)-j} \tag{4.2}$$

then

$$\binom{e\text{-}A\text{-}a}{e\text{-}A\text{-}a\text{-}j} < \binom{e\text{-}A}{e\text{-}A\text{-}j} \tag{4.3}$$

then

$$\frac{(e\text{-}A\text{-}a)!}{(e\text{-}A\text{-}a\text{-}j)!(e\text{-}A\text{-}a(e\text{-}A\text{-}a\text{-}j))!} < \tag{4.4}$$

$$\frac{(e\text{-}A)!}{(e\text{-}A\text{-}j)!(e\text{-}A(e\text{-}A\text{-}j))!}$$

then

$$\frac{(e\text{-}A\text{-}a)(e\text{-}A\text{-}a\text{-}1)(e\text{-}A\text{-}a\text{-}2)\ldots(e\text{-}A\text{-}a\text{-}j\ 1)}{j!} <$$

$$\frac{(e\text{-}A)(e\text{-}A\text{-}1)(e\text{-}A\text{-}2)\ldots(e\text{-}A\text{-}j\ 1)}{j!} \tag{4.5}$$

There will be j number of terms in the numberator of both sides of equation (4.5), each term of which is greater than one.

Since,

$$\begin{aligned}
e\text{-}A\text{-}a &< e\text{-}A \\
e\text{-}A\text{-}a\text{-}1 &< e\text{-}A\text{-}1 \\
e\text{-}A\text{-}a\text{-}2 &< e\text{-}A\text{-}2 \\
&\vdots \\
e\text{-}A\text{-}a\text{-}j+1 &< e\text{-}A\text{-}j+1
\end{aligned}$$

Then equation (4.5) is true. From it equations (4.4), (4.3), and (4.2) follows, and the theorem is proved.

Example 4.5:

The following tables are the numerical values of the 2-terminal reliability function discussed in examples (3.1 and 2). Let the branch reliabilities be all equal and have the reliability 0.9. Columns 3,4, and 5 are the values when one branch is removed from the graph. Column 6 is the value of the 2-terminal reliability function if the specified branch has the reduced reliability as shown.

	with all branches present	without branch #1	without branch #2	without branch #3	with p_x=0.5
$q_{x,y}$	.981	.81	.90	.90	.905
$q_{y,z}$	.981	.90	.81	.90	.905
$q_{x,z}$	.981	.90	.90	.81	.905

The value of q_n .972

The ideas of section (3.3) can be used to find the value of the 2-terminal reliability function in the cases of examples (3.1 and 2). Equation (3.8) is applicable in all the cases where the reliabilities of the branches are all equal; equation (3.7) where they are not all equal.

In all of these examples there exists one 1-edged path and one 2-edged path. This means, using equation (3.8),

$$\left|Q\right| P_1 = r = p = .9 = .9$$

$$\left|Q\right| P_2 = r^2\bar{r} = p^2\bar{p} = (.9)(.9)(.1) = \underline{.081}$$

The value of $q_{x,y}$ is .981

Observe that this is the same result obtained in example (4.5).

Using equation (3.7) the case of example (3.1) where the reliability of branch one has the reduced reliability, $p_1 = 0.5$.

$$\left|Q\right| P_1 = r_1 = p_1 = .5 = .5$$

$$\left|Q\right| P_2 = r_2r_3\bar{r}_1 = p_2p_3\bar{p}_1 = (.9)(.9)(.5) = \underline{.405}$$

The value of $q_{x,y}$ is .905

Observe that this is the same value found in example (4.5). It is significant to observe that in this example, if branch two had the reduced reliability, the effect on the value of $q_{x,y}$ would not be as much. Consider example (3.1), let branch two assume the reduced branch reliability, $p_2 = .5$, and calculate the value of the 2-terminal reliability function, $q_{x,y}$, using equation (3.7).

$$\left|Q\right| P_1 = r_1 = p_1 = .9 = .9$$

$$\left|Q\right| P_2 = r_2r_3\bar{r}_1 = p_2p_3\bar{p}_1 = (.5)(.9)(.1) = \underline{.045}$$

The value of $q_{x,y}$ .945

Observe that this value is higher than that one found in the case where branch one had the reduced reliability: .945 vice .905.

5. Application of Ideas without the Aid of a Computer

Some of the ideas presented in the preceeding sections can be applied, without the benefit of a computer, in the analysis of a communication net to give rough, but fast, measurements of the properties of a net. The following subsections discuss important properties of the communication net which must be examined in the event a change in the construction of the net is required. For example, the commander of several communication systems may order that one link be removed from one system and be given to a second system. The effect on the system caused by the removal of a link can be studied qualitatively without the aid of a computer.

5.1 Branch loading.

If the graph of a communication net is simple or if sufficient time is available, all the paths of the net can be identified, and the loading of each branch can be found. Immediately this indicates that the branch which is used most often should not be given up, because all the paths using that branch are also given up. On the other hand, the lightest loaded branch may not be sacrificed in every case without additional examination of the topology of the graph. That branch could be the only connection to a terminal, and the destruction of this branch would isolate the terminal. However, in certain conceivable situations, the price of isolation of a terminal may be acceptable, for example, security reasons. The full effect of branch loading on the decision to remove a branch must be tempered by additional factors.

5.2 Path Length.

Theorem (4.2) proves that long paths are not as reliable as shorter ones. Therefore, one must place a figure of merit on the communication

path in relation to its length. The removal of either one of two branches may still enable communication between stations; however, in one case the remaining paths are longer than in the other. Then, the choice would be to remove the one that would leave the shorter path. This choice may be a poor one in light of the other factors which must be considered.

5.3 Branch Reliability.

Naturally the designer of a communication net will want to avoid the use of a branch with low reliability. And, if one link has to be removed, then that one of lowest reliability should be selected. However, this is not always true, especially when that branch is the only link to a certain terminal. Again the price of isolation of that terminal may be acceptable in a situation where security considerations are paramount. Likewise, the link of highest reliability may be chosen to be removed from the given net for use elsewhere, especially in cases where the branch loading is low and there exist other paths of communication of acceptable reliability.

5.4 Secondary Path Terms.

The importance of the secondary path product terms has been stressed before. It is pointed out in this section that since they constitute the majority of terms in the 2-terminal reliability function, and they are inversely proportional in number to the length of path generating them, the loss of a short communication path is very costly in terms of reliability.

The number of k-edged paths which will be lost when a given branch, b_i, is removed can be predicted in certain cases.

Theorem 5.1:

When branch, b_i, is removed from a graph, one 1-edged path will be removed.

Proof:

There can be only one 1-edged path between terminals $\underline{x}$ and $\underline{y}$ consisting of the branch b_i alone. Removal of the branch, b_i, results in the removal of that path.

Theorem 5.2:

Let D_x and D_y be the degree of terminals $\underline{x}$ and $\underline{y}$ respectively. Let b_i be a branch connecting terminals $\underline{x}$ and $\underline{y}$. Let $\underline{c}$ be the number of additional branches connecting terminals $\underline{x}$ and $\underline{y}$. Then the number of 2-edged paths between any pair of terminals, but using branch b_i, that will be removed if branch b_i is removed, is given by

$$N_2 = (D_x + D_y - 2c - 2) \qquad (5.1)$$

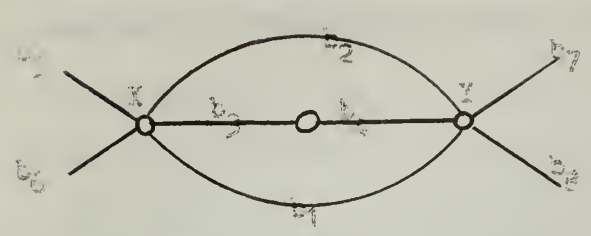
Proof:

The degree of a terminal is the number of edges incident to it. A 2-edged path using the branch, b_i, must either start at terminal $\underline{x}$ and proceed through $\underline{y}$, or start at terminal $\underline{y}$ and proceed through $\underline{x}$. In the former case, there are (D_y-1) number of possible 2-edged paths minus the $\underline{c}$ number which would form a circuit, since they are connected to terminal $\underline{x}$ also. Similarly in the latter case, there are $[(D_x-1)-c]$ number possible. The sum is N_2:

$$N_2 = D_y - 1 - c + D_x - 1 - c$$

$$N_2 = D_y + D_x - 2c - 2$$

Example 5.1



$D_x = 5$

$D_y = 5$

$c = 1$, namely b_2

Figure 5.1

$N_2 = 5 + 5 - 2(1) - 2 = 6$

Namely,

path sets $\{b_1\ b_3\}$, $\{b_1\ b_5\}$, $\{b_1 b_6\}$, $\{b_1\ b_4\}$, $\{b_1 b_7\}$,

and $\{b_1\ \ b_8\}$

Rule 5.1.

A rule which establishes a series of operations that indicates the number of 3-edged paths between any pair of terminals containing the branch, b_i, that will be removed if the branch, b_2, is removed. It is stated here without proof; however an example is given.

Step #1 Starting at node x, proceed to the terminal of a branch incident to node x.

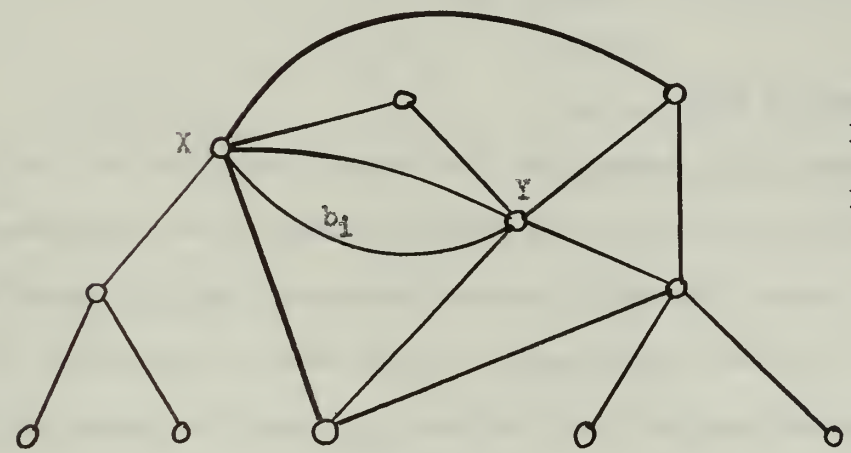
Step #2 Call this node #1 having degree D_1. Let c be the number of branches having terminals x and y, but not being branches, b_i. Let d_i be the number of branches having node y and node i as terminals. Then

$$D_i + D_y - 2 - c - 2d_i = N_3 \qquad (5.2)$$

Step #3 Proceed to all other nodes of the branches incident to node x, except the node y, of course. Repeat the calculation of equation (5.2), adding the result to the previous value of N_3.

Step #4 Proceed to the terminals of the branches incident to node y and whose terminal nodes are not those previously numbered or node x. Count the number of branches which are not incident to x or y or the nodes previously numbered in steps (2) or (3). Add this number of the previous value of N_3. N_3 is the number of 3-edged paths that will be removed if branch, b_i, is removed.

Example 5.2:



E = 16

N = 11

Figure 5.2

c = 1 $D_x = 6$

$D_1 = 3$, $d_1 = 0$

N_3 $3 + 6 - 2 - 1 - 0 = \underline{6}$

$D_2 = 3$, $d_2 = 1$

$3 + 6 - 2 - 1 - 2(1) = \underline{4}$

N_3 $N_3 = 6 + 4 = \underline{10}$

$D_3 = 2, \qquad d_3 = 1$

$2 + 6 - 2 - 1 - 2(1) = 3$

N_3 $\qquad N_3 = 10 + 3 = \underline{\underline{13}}$

$D_4 = 3 \quad , \quad d_4 = 1$

$3 + 6 - 2 - 1 - 2(1) = 4$

N_3 $\qquad N_3 = 13 + 4 = \underline{\underline{17}}$

Next node 5 where 2 branches qualify

$N_3 = 17 + 2 = \underline{\underline{19}}$

ANS: $\underline{\underline{19}}$

5.5 Number of Paths

Theorem (3.3) proves that the primary path term will not be a duplicate of any entry in the Table of Combinations regardless of the number of paths that exist between terminals $\underline{x}$ and $\underline{y}$. Therefore, each path will contribute some value, in the sense of definition (4.2), to the 2-terminal reliability function, $q_{x,y}$, and its removal will deminish this value.

An examination of the paths between terminals $\underline{x}$ and $\underline{y}$ alone is not sufficient when n-terminal reliability is concerned. What may appear to be an expendable path between terminals $\underline{x}$ and $\underline{y}$, may be a vital path between two other terminals.

While a large number of paths between the terminals of consideration indicates that the loss of one path would not cause isolation, the length of the paths may be such as to result in low reliability. All the factors mentioned previously continue to apply even when there are a large number of paths. This is also true when there are a small number of paths.

5.6 Addition of a branch.

The addition of one branch to a graph is equivalent to the addition of one communication link to a net. This can be viewed as a design problem in which there are eight cases to be studied:

What should the second branch terminal be, if the given objective is to:

		the first branch terminal is required to be:
maximize $q_{x,y}$	(1)	terminal $\underline{x}$
	(2)	terminal $\underline{a}$, not $\underline{x}$ or $\underline{y}$
	(3)	optional
maximize q_k	(4)	terminal $\underline{a}$ in set $\underline{k}$
	(5)	terminal $\underline{a}$ not in set $\underline{k}$
	(6)	optional
maximize q_n	(7)	terminal $\underline{x}$
	(8)	optional

These eight cases can lead to many more if the reliability of the branch to be added and the reliabilities of the other branches are permitted to change. The following discussion of the above cases assumes that all branch reliabilities are the same, and equal p.

Case (1):

Corollary (4.1) specifies terminal $\underline{y}$ as the second branch terminal.

Case (2):

Find all the paths between terminals $\underline{x}$ and $\underline{a}$, and $\underline{y}$ and $\underline{a}$. Choose terminal $\underline{x}$ or $\underline{y}$ depending on which has the fewer paths to terminal $\underline{a}$. This will provide the most paths, with the fewest branches, between $\underline{x}$ and $\underline{y}$.

Case (3):

Corollary (4.1) specifies the selection of terminals x and y.

Case (4):

Find all the paths between all the pairs of terminals of set k. If any one pair has an unusually small number, examine the topology of the graph in light of sections (5.1-4). Base the selection on the unusual topology. Do the same thing in the event one pair has an unusually large number of paths between them. Otherwise, locate the terminal that has the most number of paths to terminal a. Select this as the second terminal for the branch to be added. This choice will add one 1-edged path, but also the maximum number of 2- and 3-edged paths between the other terminals in the set k. This will increase the value of q_k the most.

Case (5):

Find all the paths between all the pairs of terminals of set k and terminal a. Examine the topology of the graph in light of sections (5.1-5). Especially, look for vertices of degree one or two. Unusual configurations must be solved in this manner. Otherwise, select as the second terminal that one in set k that has the largest number of paths to terminal a. See Case (4).

Case (6):

Find all the paths between all the terminal pairs in set k. Choose as the terminals of the branch to be added those between which there already exist the largest number of paths. This will not be true if an examination of the topology reveals an unusual nature to the net. See Case (4).

Case (7):

Find all the paths between all the terminals of the graph and terminal x. Examine the topology of the graph. If nothing extradorinary

appears, choose as the second terminal for the branch to be added that terminal which already has the most paths to terminal x.

Case (8):

Find all the paths between all the terminal pairs of the graph. Study the topology of the net. Choose the terminal pair between which there already exist the largest number of paths.

If there exist a large number of paths between two terminals of a graph, the paths are usually of long length. This means that they rely on many other branches, and so are of low reliability. The adding of a 1-edged path between these terminals immediately increases the 2-terminal reliability between them, and also provides an increase in the number of 2-edged paths between other pairs of terminals. In fact, it increases the number of all k-edged paths. Since there were all those paths in existence, the accessibility of the new branch by the other terminals produces a maximum number of paths between all terminals.

Lemma (3.1) shows that an isolated terminal, I, will cause both the 2- and the n-terminal reliability function to be zero. When an isolated terminal exists, clearly the topology of the graph is unusual. The isolated terminal must be selected as the second terminal for the branch to be added in each case that the terminal I is in the set k. This statement covers all the cases since the set k includes all situations from 2 to n number of terminals.

Example 5.3:

Three illustrations of a 2-terminal reliability function are discussed for the graph of figure (5.3): q_{af}, $q_{b,g}$, $q_{c,g}$. All the branch reliabilities are equal, p = .9.

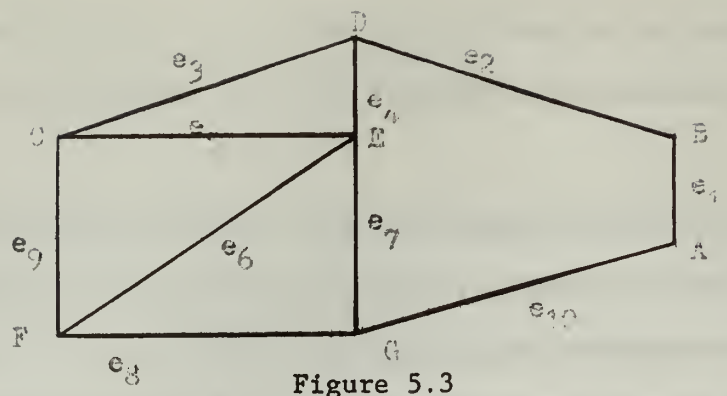


Figure 5.3

	with all branches present	without branches #3	without branches #6	without branches #9
$q_{a,f}$	.9859	.9685	.9755	.9750
$q_{b,g}$	.9940	.9621	.9922	.9897
$q_{c,g}$	.9963	.9854	.9937	.9830

Branch Loading:

	Branch number:	1	2	3	4	5	6	7	8	9	10	
$q_{a,f}$	Branch load:	6	4	4	4	4	3	5	3	4	4	10 paths
$q_{a,g}$		1	3	4	3	3	3	3	4	3	1	8 paths
$q_{c,g}$		3	4	3	4	3	4	3	3	3	3	9 paths
	Total loading:	10	11	11	11	10	10	11	10	10	8	

In the illustrations given, branch #3 is not the one to be removed. However, branch #10 should not be removed either because this would cause node A to rely completely on branch #1 for communication. The low level of branch loading for branch #10 indicated that it is ineffectively connected, not that it can be sacrificed.

6. Digital Computer Program.

6.1 Program Specifications.

A digital computer program has been written in the FORTRAN IV Language suitable for use on the IBM System 360 computer complex. The primary purpose of the program is to compute the 2-terminal reliability function, $q_{x,y}$. This program is included as Appendix I . The functional block diagram, figures (6.1), (6.2) and (6.3) is fully explained in section (6.2).

The capabilities of the program go beyond the computation of the 2-terminal reliability function, $q_{x,y}$. After finding $q_{x,y}$, the program can calculate its numerical value; then, it can recalculate it under the condition that a branch of the graph has been removed, and that a branch has a different value of reliability. (See options, Section 6.1.1.) After the second and any succeeding runs, (a run is defined as the computation of one 2-terminal reliability function,) q_k, where k is the set of terminal pairs of the preceeding runs, can be computed, and its numerical value found. In finding $q_{x,y}$, all path sets between terminal pairs are found; this information is given as a preliminary output.

6.1.1 Input

The input required by the computer program consists of three types:

(1) The number of edges, and the number of vertices of the graph are the first data required. The number of fundamental circuits is calculated from this data using the relation:

$$(e - n + 1) = w$$

Also on the first data card, the option SEL1 is provided which, if used, enables the branch reliabilities to be input. This will allow the

calculation of the numerical value of $q_{x,y}$, and later, the numerical value of q_k, if requested.

(2) The w number of fundamental circuits are the next input data. If the branch reliabilities are to be used, they are read in next.

(3) The data required by each run is read next. This data consists of three main parts:

a) the terminal vertices (numbered);

b) one path between these vertices;

c) option selections:

1) SEL - additional runs will follow

2) SEL2 - find q_k

3) SEL4 - calculate the numerical value

4) SEL5 - of $q_{x,y}$ with the specified branch

5) SEL6 - having been removed from the graph

6) SEL7 - calculate $q_{x,y}$ using the specified value of branch reliability for the specified branch

Precise details of the format for the data cards are provided in section (6.3).

6.1.2 Output.

The output provided by the computer program is fully labelled for easy reading. All the input data is printed with the generated data to provide a complete record of the run. The output of generated data include:

1) Table of Combinations;

2) path matrix;

3) branch loading information;

4) number of k-edged paths;

5) numerical values of $q_{x,y}$ and q_k.

At the conclusion of the computer program, a summary of the branch loading and the number of k-edged paths is printed. A message explaining the error stop is printed as necessary.

6.1.3 Theoretical Limitations.

In its present form, the computer program is limited to a graph containing no more than 15 edges, or 10 nodes, or 10 fundamental circuits. There must be at least one circuit in the graph. Other limitations include a maximum of 4000 entries in the Table of Combinations for all runs, and a maximum of 210 runs. The reason for these limits is the storage requirement, especially that for the Table of Combinations. Different programming techniques might be able to increase these limits slightly. The fact remains that the powers of 2 increase very rapidly, regardless of the programming technique.

6.2 Block Diagram.

The functional block diagram, figures (6.1,2 and 3), is presented to give a better description of the operation of the computer program. The blocks are identified in the listing of the computer program, Appendix I.

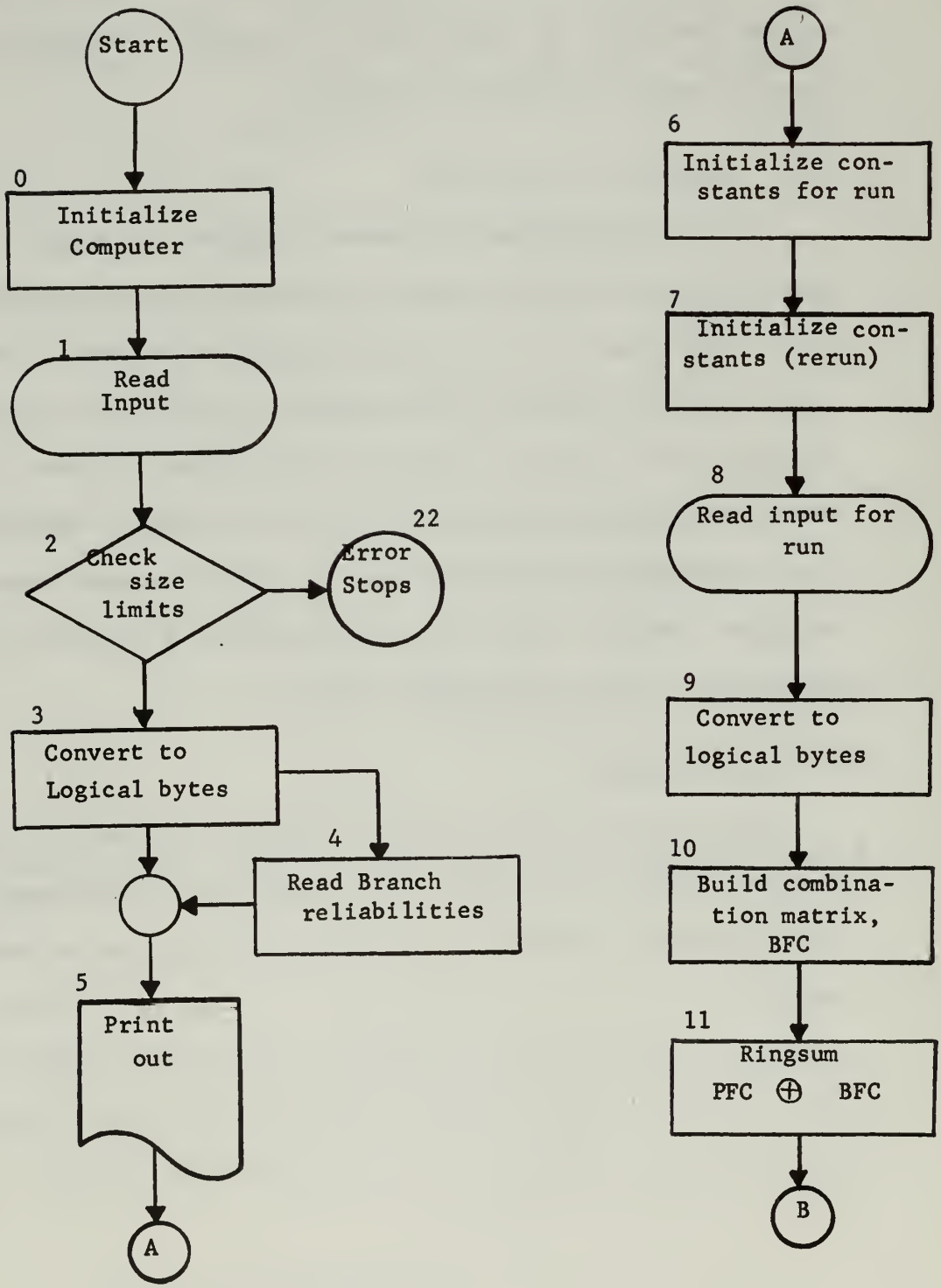

Start
0
Initialize Computer
1
Read Input
2
Check size limits
22
Error Stops
3
Convert to Logical bytes
4
Read Branch reliabilities
5
Print out
A
A
6
Initialize constants for run
7
Initialize constants (rerun)
8
Read input for run
9
Convert to logical bytes
10
Build combination matrix, BFC
11
Ringsum
PFC ⊕ BFC
B

Figure 6.1

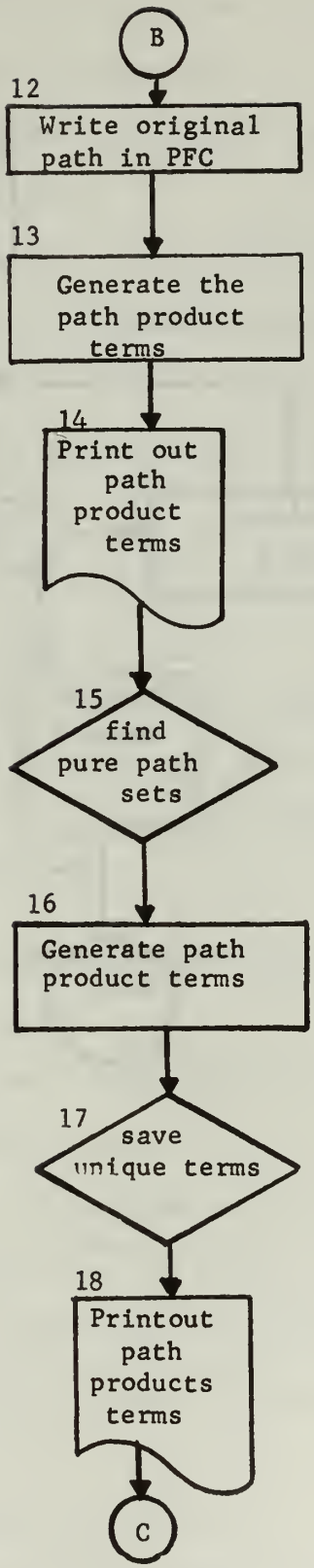

B
12
Write original path in PFC
13
Generate the path product terms
14
Print out path product terms
15
find pure path sets
16
Generate path product terms
17
save unique terms
18
Printout path products terms
C

Figure 6.2

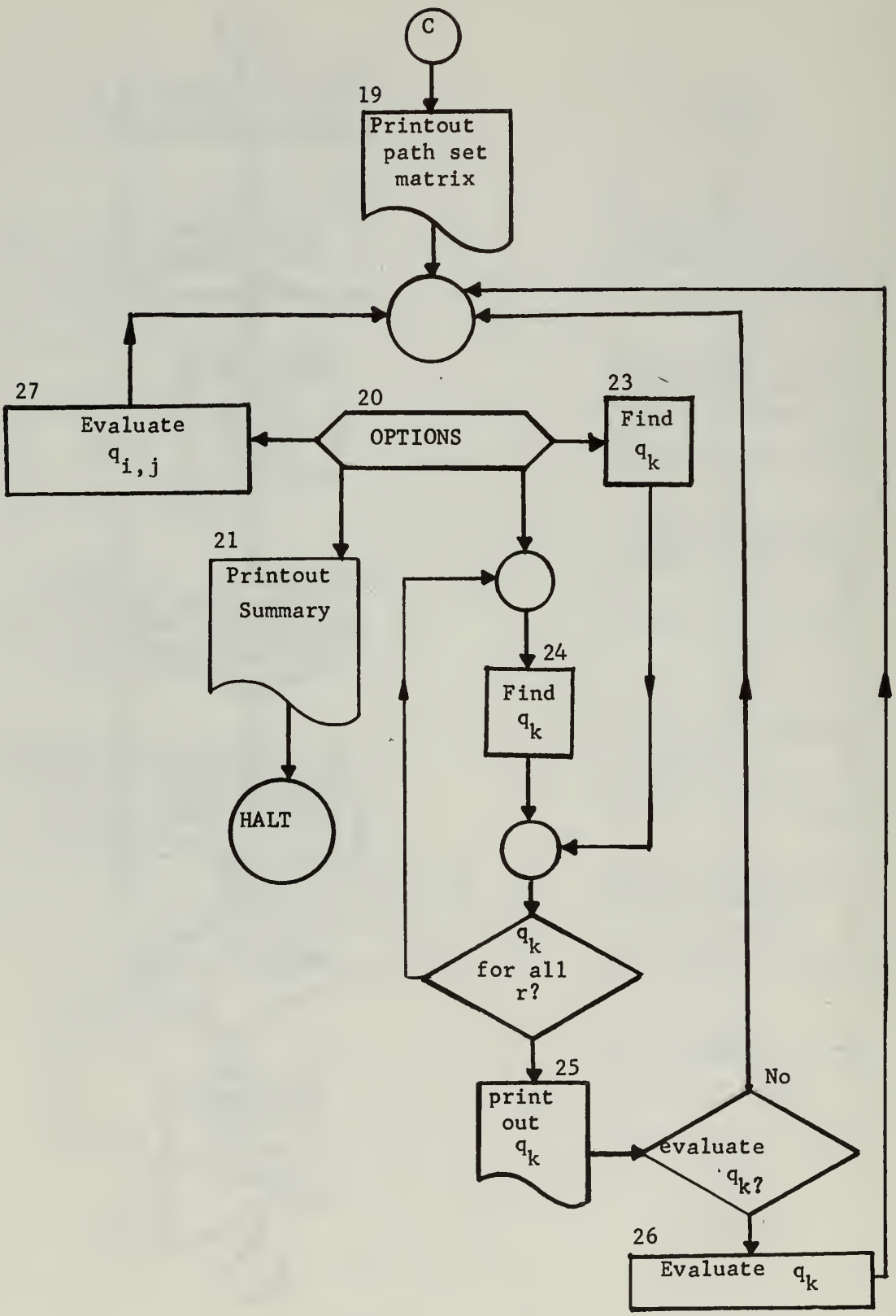

C
19
Printout path set matrix
27
Evaluate $q_{i,j}$
20
OPTIONS
23
Find q_k
21
Printout Summary
HALT
24
Find q_k
q_k for all r?
25
print out q_k
evaluate q_k?
No
26
Evaluate q_k

Figure 6.3

Description of Block Diagram

Block 0: Control cards and dimension and format statements are required to prepare the computer for running.

Block 1: Input data are read including number of branches and nodes, and the fundamental circuits.

Block 2: Graph size is tested to be within program limits.

Block 3: Input data is converted to logical elements to facilitate faster calculation.

Block 4: If branch reliabilities are to be used, they are read in now into DREL matrix.

Block 5: Preliminary printouts are made for more complete data reference for the final output.

Block 6: Values required for first run of the program are set.

Block 7: Values required for succeeding runs are set.

Block 8: Data required for the run is read including terminals and path.

Block 9: Path data is converted to logical elements, and a storage area prepared for later use.

Block 10: All possible combinations of the given circuits are generated and stored in matrix BFC.

Block 11: The ringsum operation is performed on the given path and each entry in matrix BFC, and the result is written in BRP.

Block 12: The original path is written into the path matrix, PFC.

Block 13: The primary and all the secondary path product terms are generated into OPC.

Block 14: The path product terms are printed out.

Block 15: Matrix BRP is searched for pure path sets.

Block 16: The primary and secondary path product terms are generated for the newly formed path set, and written into QTEMP.

Block 17: Each entry in QTEMP is compared to all entries in QPC. If a duplicate is not found, the set from QTEMP is written in QPC.

Block 18: Those sets added to QPC are printed out. The number of entries in QPC and the number in PFC are printed.

Block 19: The path matrix, PFC, is printed out.

Block 20: The various OPTIONS are examined. If $Q_{i,j}$ or Q_n are to be found, transfer is made; if another run it to be made, transfer is accomplished.

Block 21: Before stopping, a summary is printed out including branch loading, number of paths between each node, and path length.

Block 22: Error stops are provided.

Block 23: Generate matrix Q_n from path product terms common to the first 2 $q_{i,v}$ matrices.

Block 24: Generate matrix Q_m from the previous Q_n and the newly formed $q_{i,j}$ for this run.

Block 25: Print out matrix Q_n.

Block 26: If the branch reliabilities are provided, determine the value of Q_n. Transfer control to block 20 to check options.

Block 27: If the branch reliabilities are provided, determine the value of $q_{i,j}$. Examine the options SEL4, SEL5, SEL6, and determine the value of $q_{i,j}$ without the specified branch J. Examine option SEL7, and compute the value of $q_{i,j}$ with the revised value of branch reliability. Transfer control to block 20 to check other options.

6.3 Data Card Configuration.

The following is a description of the data cards required for the running of the computer program as described in section (6).

Data Card Type Number:	Columns:	Description and specifications:
1	1 - 3	Number of edges, right justified; max: 15
	4 - 6	Number of nodes, right justified; max: 10
	7 - 9	If 0 or Blank: no branch reliabilities are to be used; if nonzero: branch reliabilities are to be read and used.
	10 - 72	Disregarded.
2	1 - 15	Fundamental circuit set: Up to 15 digits, 0 or 1, left justified, representing the edges in the circuit; 1 if edge present; 0 if edge not present
	15 - 72	Disregarded.
3	1 - 5	Blank
	6 - 13	Decimal point and 7 digits to represent
	14 - 21	the branch reliability of the succes-
	22 - 29	sive branches 1, 2, . . . , e
	30 - 37	One field for each edge. Up to five
	38 - 45	edges per card.
	46 - 72	Disregarded.

Type Number:	Columns:	Description and Specifications:
4	1 - 3	Identifying number of starting node
	4 - 6	Identifying number of terminal node
	7 - 9	if 0, no more cards will be read; if not 0, additional cards read.
	10 - 12	if nonzero, q_k will be computed using the previously found $q_{i,j}$'s from all previous runs as the set $\underline{k}$; if 0, then this is not done.
	13 - 27	left justified, up to 15 digits, 0 to 1, representing the edges in a path between the subject terminals; 0 if edge not present 1 if edge present only e number of digits processed
	28 - 30	an integer, less than or equal to e, representing an edge that has been removed from the graph. A value for $q_{i,j}$ is found assuming that edge is removed from the graph.
	31 - 33	another edge number processed as that on in the preceeding field.
	34 - 37	Same as above
	38 - 40	An edge number whose value of branch reliability it is desired to change temporarily, and recompute $q_{i,j}$ with the new value of reliability.

Type Number:	Columns:	Description and Specifications:
	41 - 48	A decimal point and 7 digits represent- the branch reliability of the edge specified in the preceeding field.

7. Conclusions.

This work shows that the reliability function as defined is an effective instrument in the analysis and design of communication nets. The theorems that are developed can be applied in many cases without the use of a digital computer to provide direction in the modification of a net. The computer can provide fast data to quantify the effect of any change to a communication net with respect to k-terminal reliability, as the results of the computer study indicate.

The computer program can be modified by specialization to facilitate sensitivity studies for branch removal or branch addition. In so doing, the limitations on graph construction can be modified. Moreover, equation (3.7) can be used in a computer program to study the 2-terminal reliability function alone.

BIBLIOGRAPHY

1. C. B. Allendoerfer and C. O. Oakley, Fundamentals of Freshman Mathematics, McGraw-Hill Book Company, Inc., New York, 1959.

2. S. G. Chan and S. P. Chan, "Some Applications of Topology to Probabilistic Communication Networks", Ninth Midwest Symposium on Circuit Theory, Oklahoma State University, Stillwater, Oklahoma, May 9-10, 1966.

3. Y. Fu, "Applications of Topological Methods to Probabilistic Communication Networks", IEEE TRANS on Communication Technology, Vol., COM-13, No. 3, pp 301-307, September 1965.

4. Y. Fu and S. S. Yau, "A Note on the Reliability of Communications Network", J. SIAM, Vol. 10, pp 469-474, September, 1962.

5. S. Seshu and M. B. Reed, Linear Graphs and Electrical Networks, Reading, Mass., Addison-Wesley, 1961.

6. O. Wing and D. Demetrious, "Analysis of Probabilistic Networks", IEEE TRANS on Communication Technology, Vol. COM-12, pp 28-40, September, 1964.

APPENDIX I

The digital computer program, as described in section (6) is presented as Appendix I. The programming language is FORTRAN IV and the program is suitable for use on the IBM System 360 computer complex.

```
C
C      BLOCK  NUMBER  0
C
0001        IMPLICIT LOGICAL*1(A-C,O-Q),
           1INTEGER*4(E-N,R-Z)
0002        DIMENSION XBF(10,15),BCM(10,15),
           1BFC(1024,15),BRP(1024,15),PFC(1024,15),
           1XPFC(210,15),XY(210,15),ENO(210,15),
           1GNO(210),QNO(400,15),QN1(400,15),
           1QTS(15),QPC(4000,15),QTEMP(400,15),
           1UNO(210), URUN(210),XPTH(15),
           1PTS(15),ST(210),SP(210),DREL(15)
C
C
0003       45 FORMAT(/,26H  THE VALUE OF Q-SUB-(I,J)
             115H WITHOUT BRANCH1I3)
0004       46 FORMAT(/,26H  THE VALUE OF Q-SUB-(I,J)
             114H WITH BRANCH  1I3,/,
             125H HAVING RELIABILITY VALUE1F8.6)
0005       76 FORMAT(/,6H  STOP1I5)
0006       77 FORMAT(/,22H  TOTAL BRANCH LOADING)
0007       78 FORMAT(/,25H  IN PAIRS, THE NUMBER OF
             133H K-EDGED PATHS AND BRANCH LOADING)
0008       79 FORMAT(1H1,25H  THE TOTAL NUMBER OF PAT
             120HHS FOUND IN EACH RUN)
0009       80 FORMAT(3I3)
0010       81 FORMAT(15I1)
0011       82 FORMAT(2I6,15I1,1I2)
0012       83 FORMAT(5X,15I3)
0013       84 FORMAT(1H1,20H  THE GIVEN PATH WAS)
0014       85 FORMAT(5X,/)
0015       86 FORMAT(1H1,10H THERE ARE 1I5,
             110H LINKS AND1I5,6H NODES/)
0016       87 FORMAT(/,25H  EDGE 1  2  3  4  5  6
             125H7  8  9 10 11 12 13 14 15/)
0017       88 FORMAT(/,29H  THE GIVEN CIRCUIT MATRIX IS)
0018       89 FORMAT (/,20H  THE PATH MATRIX IS)
0019       90 FORMAT(/,5H   R=1I3,5H   U=1I3)
0020       91 FORMAT(/,20H  MATRIX QPC FOLLOWS)
0021       92 FORMAT(4I3,15I1,4I3,1F8.6)
0022       93 FORMAT(5X,5F8.6)
0023       94 FORMAT(/,30H  THE BRANCH RELIABILITIES ARE)
0024       95 FORMAT(/,13H  Q-SUB-I,J =1F10.7)
0025       96 FORMAT(/,11H  Q-SUB-K =1F10.7)
0026       97 FORMAT(/,19H  MATRIX QK FOLLOWS)
0027       98 FORMAT(1H1,8H RUN NO.1I6,
             16H  FROM1I5,4H  TO1I5,/)
0028       99 FORMAT(/,10H THERE ARE1I5,6H PATHS)
```

```
C
C       BLOCK  NUMBER  1,2,3,4, AND 5
C
0029            READ(5,80) E,N,SEL1
0030            WRITE(6,86) E,N
0031            W=E-N+1
0032            IF(E.GT.15)  GO TO 3
0033            IF(E.LE. 0)  GO TO 3
0034            IF(N.GT.10)  GO TO 3
0035            IF(N.LE. 0)  GO TO 3
0036            IF(W.GT.10)  GO TO 3
0037            IF(W.LE. 0)  GO TO 3
0038            DO 71 I=1,W
0039         71 READ(5,81)(XBF(I,J),J=1,E)
0040            DO 75 I=1,W
0041            DO 70 J=1,E
0042            IF (XBF(I,J).EQ.0)  BCM(I,J)=.FALSE.
0043            IF(XBF(I,J).EQ.1)  BCM(I,J)=.TRUE.
0044         70 CONTINUE
0045         75 CONTINUE
0046            WRITE(6,88)
0047            DO 72 I=1,W
0048         72 WRITE(6,83)(XBF(I,J),J=1,E)
0049            WRITE (6,85)
0050            IF(SEL1.EQ.0) GO TO 10
0051            READ(5,93)(DREL(J),J=1,E)
0052            WRITE(6,94)
0053            WRITE(6,93)(DREL(J),J=1,E)
C
C       BLOCK  NUMBER  6 AND 7
C
0054         10 TIMS=0
0055            OPT = .FALSE.
0056            QTAG =.FALSE.
0057            QTAG1=.FALSE.
0058            QTAG2=.FALSE.
0059            GR=0
0060            R=0
0061            U=0
0062          1 R=R+1
0063            URAN=U+1
0064            U=0
0065            UV=0
C
C       BLOCK  NUMBER  8 AND 9
C
0066            READ(5,92)ST(R),SP(R),SEL,SEL2,XPTH,
              1 SEL4,SEL5,SEL6,SEL7,DCHG
```

```
0067                WRITE(6,84)
0068                WRITE(6,87)
0069                WRITE(6,83) XPTH
0070                WRITE (6,85)
0071                DO 54 I=1,E
0072                IF (XPTH(I).EQ.0)  PTS(I)=.FALSE.
0073                IF (XPTH(I).EQ.1)  PTS(I)=.TRUE.
0074                XY(R,J)=0
0075             54 ENO(R,J)=0
                 C
                 C      BLOCK  NUMBER  10
                 C
0076                IF(OPT)GO TO 51
0077                OPT = .TRUE.
0078                DO 55 I=1,E
0079                BFC(1,I)=.FALSE.
0080             55 BFC(2,I)=BCM(1,I)
0081                K=2
0082                G=3
0083                IF(W.EQ.1)   GO TO 51
0084                DO 58 I=2,W
0085                DO 57 J=1,K
0086                DO 56 F=1,E
0087                A=BFC(J,F)
0088                B=BCM(I,F)
0089                C =.FALSE.
0090                IF (A.OR.B)  C=.TRUE.
0091                IF (A.AND.B) C=.FALSE.
0092                BFC(G,F) = C
0093             56 CONTINUE
0094                G=G+1
0095             57 CONTINUE
0096             58 K=G-1
                 C
                 C      BLOCK  NUMBER  11
                 C
0097             51 T=GR+1
0098                DO 38 I=1,K
0099                DO 37 J=1,E
0100                A=BFC(I,J)
0101             52 B=PTS(J)
0102                C =.FALSE.
0103                IF (A.OR.B)  C=.TRUE.
0104                IF (A.AND.B) C=.FALSE.
0105             37 BRP(I,J)=C
0106             38 CONTINUE
                 C
                 C      BLOCK  NUMBER  12, 13, AND 14
```

```
C
0107            U=URAN
0108            UV=U
0109            DO 103 J=1,E
0110            QTS(J)=PTS(J)
0111            QPC(U,J)=PTS(J)
0112          103 PFC(T,J)=PTS(J)
0113            DO 115 J=1,E
0114            IF(QTS(J).) GO TO 115
0115            QTS(J)=.TRUE.
0116            USTOP=UV
0117            DO 102 V=U,USTOP
0118            UV=UV+1
0119            DO 101 JJ=1,E
0120            C=.FALSE.
0121            A=QPC(V,JJ)
0122            B=QTS(JJ)
0123            IF(A.OR.B) C=.TRUE.
0124            QPC(UV,JJ)=C
0125          101 CONTINUE
0126          102 CONTINUE
0127            QTS(J)=.FALSE.
0128          115 CONTINUE
0129            U=UV
0130            WRITE(6,91)
0131            WRITE(6,87)
0132            H=0
0133            DO 225 HQ=URAN,UV
0134            H=H+1
0135            DO 224 J=1,E
0136            XPFCI(H,J)=1
0137            IF(QPC(HQ,J)) GO TO 224
0138            XPFCI(H,J)=0
0139          224 CONTINUE
0140            WRITE(6,83)(XPFCI(H,J),J=1,E)
0141          225 CONTINUE
C
C       BLOCK  NUMBER  15
C
0142            DO 228 I=1,K
0143            AD=.FALSE.
0144            DO 226 J=1,E
0145            IF(PTS(J)) GO TO 230
0146          226 CONTINUE
0147            IF(AD) GO TO 228
0148            GO TO 236
0149          230 IF(BRP(I,J)) GO TO 234
0150            GO TO 236
```

```
0151          234 AD=.TRUE.
0152              GO TO 226
              C  CIRCUIT EXISTENCE TEST
0153          236 AD=.FALSE.
0154              DO 299 L=1,K
0155              AD=.FALSE.
0156              DO 295 J=1,E
0157              IF(BFC(L,J)) GO TO 298
0158          295 CONTINUE
0159              IF (AD)  GO TO 228
0160              GO TO 300
0161          238 AD=.TRUE.
0162              GO TO 295
0163          298 IF (BRP(I,J)) GO TO 238
0164          300 CONTINUE
0165          299 CONTINUE
              C
              C       BLOCK  NUMBER  16, 17 AND 18
              C
0166              T=T+1
0167              V=1
0168              DO 104 J=1,E
0169              QTS(J)=BRP(I,J)
0170              QTEMP(1,J)=BRP(I,J)
0171          104 PFC(T,J)=BRP(I,J)
0172              DO 108 J=1,E
0173              IF(QTS(J)) GO TO 108
0174              QTS(J)=.TRUE.
0175              STEP =V
0176              DO 106 IV=1,STEP
0177              V=V+1
0178              DO 105 JJ=1,E
0179              C=.FALSE.
0180              A=QTEMP(IV,JJ)
0181              B=QTS(JJ)
0182              IF(A.OR.B)  C=.TRUE.
0183              QTEMP(V,JJ)=C
0184          105 CONTINUE
0185          106 CONTINUE
0186              QTS(J)=.FALSE.
0187          108 CONTINUE
0188              KTR=0
0189              DO 113 H=1,V
0190              DO 111 IV=URAN,U
0191              DO 110 J=1,E
0192              A=QTEMP(H,J)
0193              B=QPC(IV,J)
0194              IF(A.AND.B) GO TO 110
```

```
0195                IF(A.OR.B)   GO TO 111
0196           110  CONTINUE
0197                GO TO 113
0198           111  CONTINUE
0199                KTR=KTR+1
0200                UV=UV+1
0201                DO 112 J=1,E
0202                XPFCI(1,J)=1
0203                IF(QTEMP(H,J)) GO TO 112
0204                XPFCI(1,J)=0
0205           112  QPC(UV,J)=QTEMP(H,J)
0206                WRITE(6,83)(XPFCI(1,J),J=1,E)
0207           113  CONTINUE
0208                U=UV
0209                IF(U.GT.4000) GO TO 5
0210           228  CONTINUE
0211                TNO=T-GR
0212                WRITE(6,85)
0213                WRITE(6,99) TNO
0214                WRITE(6,89)
0215                WRITE(6,87)
                C
                C       BLOCK  NUMBER  19
                C
0216                TSTR = GR + 1
0217                H=0
0218                DO 127 I =TSTR,T
0219                H=H+1
0220                X=0
0221                DO 126 J=1,E
0222                XPFCI(H,J) =1
0223                IF(PFC(I,J)) GO TO 128
0224                XPFCI(H,J) = 0
0225           126  CONTINUE
0226                WRITE (6,83) (XPFCI(H,J),J=1,E)
0227                XY(R,X) =XY(R,X)+1
0228           127  CONTINUE
0229                WRITE(6,78)
0230                WRITE(6,87)
0231                WRITE(6,83)(XY(R,X),X=1,E)
0232                WRITE(6,83)(ENC(R,J), J=1,E)
0233                WRITE (6,85)
0234                URUN(R)=U
0235                UNO(R)=U-URAN+1
0236                GNO(R)=TNO
0237                GR=T
                C
                C       BLOCK  NUMBER  20, 21 AND 22
```

```
C
0238        IF(SEL1.NE.0) GO TO 15
0239      119 IF(QTAG)  GO TO 120
0240        QTAG=.TRUE.
0241      121 IF (SEL.EQ.0)  GO TO 53
0242        GO TO 1
0243      128 ENO(R,J)=ENO(R,J)+1
0244        X=X+1
0245        GO TO 126
0246       53 WRITE(6,79)
0247        WRITE(6,83)(GNO(I),I=1,R)
0248         WRITE(6,78)
0249        DO 6 J=1,E
0250        6 UNO(J)=0
0251        DO 2 I=1,R
0252        WRITE(6,98) I,ST(I),SP(I)
0253        DO 7 J=1,E
0254        7 UNO(J)=UNO(J)+ENO(I,J)
0255        WRITE(6,83)(XY(I,J),J=1,E)
0256        WRITE(6,83)(ENO(I,J),J=1,E)
0257        2 WRITE(6,85)
0258        WRITE(6,77)
0259        WRITE(6,83)(UNO(J),J=1,E)
0260        9 STOP
0261        3 RESTP=99
0262        GO TO 8
0263        4 RESTP=999
0264        GO TO 8
0265        5 RESTP=888
0266        8 WRITE(6,76) RESTP
0267        GO TO 9
C
C       BLOCK  NUMBER  23
C
0268      120 IF (SEL2.EQ.0)  GO TO 121
0269        IF(QTAG1)  GO TO 150
0270        QTAG1=.TRUE.
0271        Z=1
0272        ZA1=UNO(1)
0273        ZA2=ZA1+1
0274        ZA3=ZA1+UNO(2)
0275        DO 125  YZ=1,ZA1
C276        DO 124 YY=ZA2,ZA3
0277        DO 123  J=1,E
0278        A=QPC(YZ,J)
0279        B=QPC(YY,J)
0280        IF(A.AND.B)  GO TO 122
0281        IF (A.OR.B)  GO TO 124
```

```
0282            122 QNO(Z,J)=A
0283            123 CONTINUE
0284                Z=Z+1
0285            124 CONTINUE
0286            125 CONTINUE
0287                YR=2
0288                GO TO 161
                C
                C       BLOCK  NUMBER  24 AND 25
                C
0289            150 ZSTR=URUN(YR)+1
0290                YR=YR+1
0291                ZMX=Z-1
0292                ZAR=URUN(YR)
0293                Z=1
0294                DO 160 YZ=ZSTR,ZAR
0295                DO 159 YY=1,ZMX
0296                DO 158  J=1,E
0297                A=QPC(YZ,J)
0298                IF(QTAG2)  GO TO 154
0299                B=QNO(YY,J)
0300            151 IF(A.AND.B)  GO TO 152
0301                IF(A.OR.B)  GO TO 159
0302            152 IF(QTAG2)  GO TO 156
0303                QN1(Z,J)=A
0304                GO TO 158
0305            154 B=QN1(YY,J)
0306                GO TO 151
0307            156 QNO(Z,J)=A
0308                GO TO 158
0309            158 CONTINUE
0310                Z=Z+1
0311            159 CONTINUE
0312            160 CONTINUE
0313                QTAG2=.NOT.QTAG2
0314            161 IF(YR.EQ.R) GO TO 162
0315                GO TO 150
0316            162 ZMX=Z-1
0317                WRITE(6,97)
0318                DO 27 YD=1,ZMX
0319                DO 26 J=1,E
0320                XPFCI(1,J)=1
0321                IF(QTAG2) GO TO 24
0322                IF(QNO(YD,J)) GO TO 26
0323             25 XPFCI(1,J)=0
0324             26 CONTINUE
0325                WRITE(6,83)(XPFCI(1,J),J=1,E)
0326             27 CONTINUE
```

```
0327                GO TO 164
0328             24 IF(QN1(YD,J)) GO TO 26
0329                GO TO 25
                C
                C       BLOCK  NUMBER  26
                C
0330            164 IF(SEL2.NE.2) GO TO 121
0331                ZMX=Z-1
0332                DSUM=0.0
0333                DO 168 YD=1,ZMX
0334                DTEMP=1.0
0335                DO 167 J=1,E
0336                IF(QTAG2) GO TO 165
0337                IF(QNO(YD,J)) GOTO 163
0338            166 DTEMP=DTEMP*(1.0-DREL(J))
0339            167 CONTINUE
0340                DSUM=DSUM+DTEMP
0341            168 CONTINUE
0342                WRITE(6,96) DSUM
0343                GO TO 121
0344            163 DTEMP=DTEMP*DREL(J)
0345                GO TO 167
0346            165 IF(QN1(YD,J)) GO TO 163
0347                GO TO 166
                C
                C       BLOCK  NUMBER  27
                C
0348             14 DTEMP=DTEMP*DREL(J)
0349                GO TO 16
0350             15 DSUM=0.0
0351                USTP=URUN(R)
0352                DO 17 UD=URAN,USTP
0353                DTEMP=1.0
0354                DO 16 J=1,E
0355                IF(QPC(UD,J)) GO TO 14
0356                DTEMP=DTEMP*(1.0-DREL(J))
0357             16 CONTINUE
0358                DSUM=DSUM+DTEMP
0359             17 CONTINUE
0360                WRITE(6,98) R,ST(R), SP(R)
0361                WRITE(6,95) DSUM
0362                WRITE(6,85)
0363                IF(SEL4.LE.0)  GO TO 31
0364                SELX=SEL4
0365             30 IF(SELX.GT.15)  GO TO 3
0366                DTEM1=DREL(SELX)
0367                DREL(SELX)=0.0
0368                DSUM=0.0
```

```
0369                    DO 32 UD=URAN,USTP
0370                    IF(QPC(UD,SELX)) GO TO 32
0371                    DTEMP=1.0
0372                    DO 33 J=1,E
0373                    IF(QPC(UD,J)) GO TO 36
0374                    DTEMP=DTEMP*(1.0-DREL(J))
0375                 33 CONTINUE
0376                    DSUM=DSUM+DTEMP
0377                 32 CONTINUE
0378                    WRITE(6,45) SELX
0379                    WRITE(6,95) DSUM
0380                    WRITE(6,85)
0381                    DREL(SELX)=DTEM1
0382                 31 IF(SEL5.LE.0)  GO TO 34
0383                    SELX=SEL5
0384                    SEL5=0
0385                    GO TO 30
0386                 36 DTEMP=DTEMP*DREL(J)
0387                    GO TO 33
0388                 34 IF(SEL6.LE.0)  GO TO 35
0389                    SELX=SEL6
0390                    SEL6=0
0391                    GO TO 30
0392                 35 IF(SEL7.LE.0)  GO TO 119
0393                    IF(SEL7.GT.15)  GO TO 3
0394                    DTEM1=DREL(SEL7)
0395                    DREL(SEL7)=DCHG
0396                    DSUM=0.0
0397                    DO 44 UD=URAN,USTP
0398                    DTEMP=1.0
0399                    DO 43 J=1,E
0400                    IF(QPC(UD,J)) GO TO 42
0401                    DTEMP=DTEMP*(1.0-DREL(J))
0402                 43 CONTINUE
0403                    DSUM=DSUM+DTEMP
0404                 44 CONTINUE
0405                    WRITE(6,46) SEL7,DCHG
0406                    WRITE(6,95) DSUM
0407                    WRITE(6,85)
0408                    DREL(SEL7)=DTEM1
0409                    GO TO 119
0410                 42 DTEMP=DTEMP*DREL(J)
0411                    GO TO 43
                     C
                     C
0412                    END
```

```
C   GR = TOTAL NO OF PATHS AT START OF RUN
C   T IS USED AS A RUNNING INDEX
C   GNO(R) IS USED AS A TOTAL FOR THE RUN
C   IF SEL IS ZERO, COMPUTER WILL HALT AFTER
C   THIS RUN;  IF NONZERO, THEN AN ADDITIONAL
C   DATA CARD WILL BE READ AFTER THIS RUN
C   IF SEL2 IS ZERO, NO Q-SUB-K FUNCTION
C   WILL BE FOUND;  IF NONZERO, THEN Q-SUB-N
C   WILL BE DETERMINED
C   NOW COMPUTE Q-SUB-N
C   UNO(R) CONTAINS LENGHT OF Q-SUB-I,J(P)
C   R IS THE RUN NUMBER
C   ST IS THE STARTING TERMINAL
C   SP IS THE ENDING TERMINAL
C   W IS THE NO. OF INDEPENDENT CIRCUITS
C   K IS THE NO. OF POSSIBLE COMBINATIONS
C       OF THE GIVEN CIRCUITS
C   GR IS USED AS A RUN-TOTAL-ER
C   STOP 99 MEANS THE NUMBER OF NODES, EDGES, OR CIRCUITS
C   IS GREATER THAN THE LIMITS OF 15, 10, AND 10
C   STOP 999 MEANS THE LIMIT IF Q-SUB-K HAS BEEN REACHED
C   STOP 888 MEANS TOO MANY TERMS HAVE BEEN PRODUCED
C   AND ALL RESULTS ARE NOW IN DOUBT
C   SEL IS THE SELECTION CODE
C   IF SEL1 = 0, NO BRANCH RELIABILITIES ARE READ
C   IF SEL1 NOT ZERO, THE VALUE FOR Q-SUBI,J IS FOUND
C   IF SEL2 = 2,  THE VALUE FOR Q-SUB-N IS FOUND
C   MATRIX XBF CONTAINS THE INDEPENDENT CIRCUITS
C   MATRIX BCM IS XBF AS LOGICAL ELEMENTS
C   MATRIX XPTH CONTAINS THE GIVEN PATH SET
C   MATRIX BFC CONTAINS ALL POSSIBLE COMBIN-
C   ATIONS OF BCM
C   MATRIX BRP CONTAINS THE RINGSUM OF THE
C   PATH AND BFC
C   MATRIX PFC IS THE PATH MATRIX, INDEXED
C   BY T OVER ALL RUNS
C   MATRIX XPFCI IS THE INTEGER VERSION OF PFC
C   MATRIX XY COUNTS THE NUMBER OF K-EDGED PATHS
C   MATRIX ENO CONTAINS BRANCH LOADING
C   MATRIX GNO HAS THE NUMBER OF PATHS
C   MATRIX
C   THIS ADDS UNIQUE PATH PRODUCT TERMS
C   TO THE COLLECTION NOW IN QPC
C   BUILD THE PATH PRODUCT TERMS
C   STORE FOR LATER TESTING OF UNIQUENESS
```

APPENDIX II

The problems solved in examples (3.1-3) and (4.5) are combined and solved by the digital computer program. The output of the running of the program is included as Appendix II.

THERE ARE 3 LINKS AND 3 NODES

THE GIVEN CIRCUIT MATRIX IS
1 1 1

THE BRANCH RELIABILITIES ARE
0.9000000.9000000.900000

```
 THE GIVEN PATH WAS
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     1  0  0  0  0  0  0  0  0  0  0  0  0  0  0

 MATRIX QPC FOLLOWS
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     1  0  0
     1  1  0
     1  0  1
     1  1  1
     0  1  1

THERE ARE    2 PATHS
 THE PATH MATRIX IS
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     1  0  0
     0  1  1
 IN PAIRS, THE NUMBER OF K-EDGED PATHS AND BRANCH LOADING
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     1  1  0
     1  1  1
```

```
RUN NO.      1  FROM     1  TO     2

Q-SUB-I,J = C.9809998

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  1
Q-SUB-I,J = 0.8099999

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  2
Q-SUB-I,J = C.8999999

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  3
Q-SUB-I,J = 0.8999999

THE VALUE OF Q-SUB-(I,J) WITH BRANCH     1
HAVING RELIABILITY VALUE0.500000
Q-SUB-I,J = 0.9049999
```

THE GIVEN PATH WAS

EDGE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

MATRIX QPC FOLLOWS

EDGE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	0												
	1	1	0												
	0	1	1												
	1	1	1												
	1	0	1												

THERE ARE 2 PATHS

THE PATH MATRIX IS

EDGE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	0												
	1	0	1												

IN PAIRS, THE NUMBER OF K-EDGED PATHS AND BRANCH LOADING

EDGE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	1	0												
	1	1	1												

```
RUN NO.      2  FROM     1  TO     3

Q-SUB-I,J = 0.9909998

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  1
Q-SUB-I,J = 0.8999999

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  2
Q-SUB-I,J = 0.8099999

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  3
Q-SUB-I,J = 0.3999999

THE VALUE OF Q-SUB-(I,J) WITH BRANCH     2
HAVING RELIABILITY VALUE0.500000
Q-SUB-I,J = 0.9049999

MATRIX QK FOLLOWS
      1   1   0
      1   0   1
      1   1   1
      0   1   1

Q-SUB-K = 0.9719998
```

```
 THE GIVEN PATH WAS
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     0  0  1  0  0  0  0  0  0  0  0  0  0  0  0

 MATRIX QPC FOLLOWS
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     0  0  1
     1  0  1
     0  1  1
     1  1  1
     1  1  0

THERE ARE    2 PATHS
 THE PATH MATRIX IS
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     0  0  1
     1  1  0
 IN PAIRS, THE NUMBER OF K-EDGED PATHS AND BRANCH LOADING
EDGE 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
     1  1  0
     1  1  1
```

```
RUN NO.      3  FROM    2  TO     3

Q-SUB-I,J = 0.9809998

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  1
Q-SUB-I,J = 0.8999999

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  2
Q-SUB-I,J = 0.8999999

THE VALUE OF Q-SUB-(I,J) WITHOUT BRANCH  3
Q-SUB-I,J = 0.8099999

THE VALUE OF Q-SUB-(I,J) WITH BRANCH     3
HAVING RELIABILITY VALUE0.500000
Q-SUB-I,J = 0.9049999

MATRIX OK FOLLOWS
     1  0  1
     0  1  1
     1  1  1
     1  1  0

Q-SUB-K = 0.9719998
```

THE TOTAL NUMBER OF PATHS FOUND IN EACH RUN
2 2 2

IN PAIRS, THE NUMBER OF K-EDGED PATHS AND BRANCH LOADING

RUN NO. 1 FROM 1 TO 2

1 1 0
1 1 1

RUN NO. 2 FROM 1 TO 3

1 1 0
1 1 1

RUN NO. 3 FROM 2 TO 3

1 1 0
1 1 1

TOTAL BRANCH LOADING
3 3 3

INITIAL DISTRIBUTION LIST

		No. copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, Calif. 93940	2
3.	Professor S. G. Chan Department of Electrical Engineering Naval Postgraduate School Monterey, Calif. 93940	6
4.	LT R. J. Huber, USN U. S. Naval Base FPO New York, N. Y. 09593	1
5.	Prof. S. P. Chan Department of Electrical Engineering University of Santa Clara Santa Clara, Calif. 95053	1
6.	Mr. Thomas Fagan General Electric Company Advanced Mission Requirements Spacecraft Division - P.O. Box 8555 Room U4633 VFSTC Philadelphia, Pa. 19101	2

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
The Analysis and Design of Communication Nets Using Reliability Functions

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (Last name, first name, initial)
HUBER, Raymond James

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1967	86	6

8a. CONTRACT OR GRANT NO. b. PROJECT NO. N/A c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) N/A 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES
[redacted]

11. SUPPLEMENTARY NOTES N/A	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

Topological methods may be employed in an effective manner in the analysis and design of a communication net. One application is in the determination of the reliability of the net with respect to both complete communication and k-terminal communication. The effect on this reliability caused by the removal of one link is studied.

The k-terminal reliability function is defined and its application to analysis and design of communication nets is demonstrated. A digital computer program is presented, and examples of its use are included. A formula is given to find the value of the 2-terminal function.

DD FORM 1 JAN 64 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Communication						
Reliability						
Topology						
Nets						